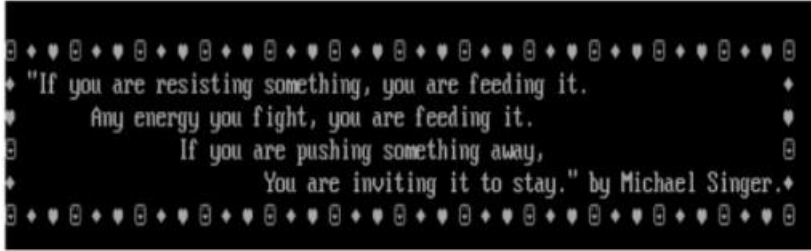
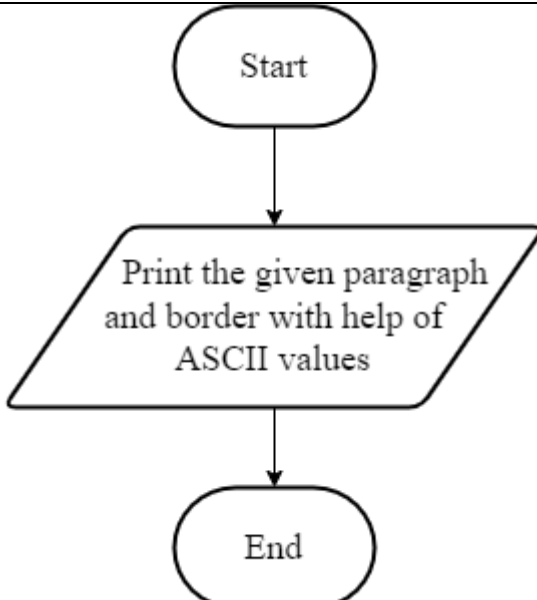


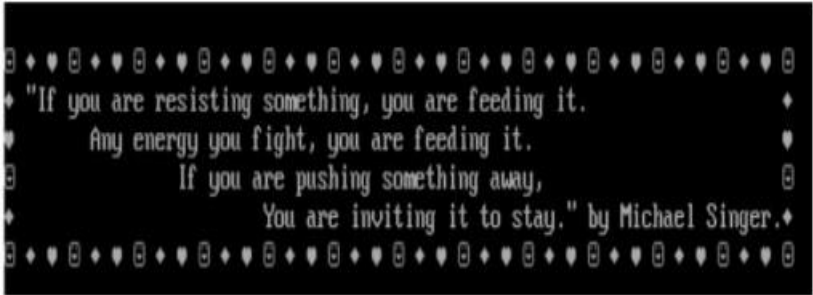
CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY
CHANDUBHAI S. PATEL INSTITUTE OF TECHNOLOGY

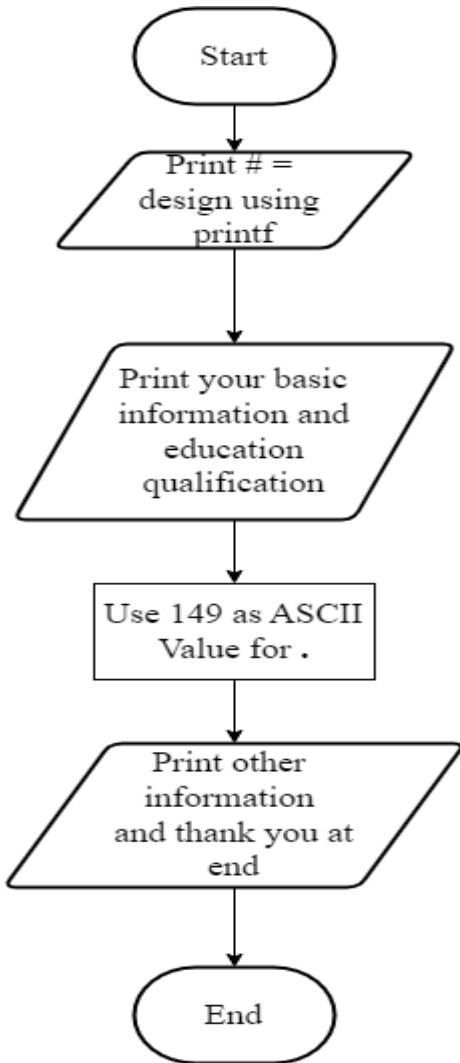
Subject Code and Name: CE143 Computer Concepts and Programming

Semester: I

Academic year: 2022-2023

NO	Aim of the practical
	SET-1:Introduction to 'C' Language
Practical 1.1	<p>Write a C program that will output this passage by Michael Singer. Make sure your output looks exactly as shown here (including spacing, line breaks, punctuation, and the title and author). Use Required Escape Sequence and ASCII Value. Outcome:</p>  <p>Note: There are three shapes in the output: Smiling Face, Diamond & Heart. The ASCII Value for Smiling face is 1. The ASCII Value for Diamond is 4. The ASCII Value for Heart is 3. Also draw flowchart and write algorithm. Try this example on Turbo C or Code blocks only.</p>
Algorithm	<p>Step 1: Start.</p> <p>Step 2: Find the ASCII values for smiling face, diamond and heart.</p> <p>Step 3: Print the given paragraph with proper spacing line break and punctuation.</p> <p>Step 4: End.</p>
Flowchart	 <pre> graph TD Start([Start]) --> Print[/Print the given paragraph and border with help of ASCII values/] Print --> End([End]) </pre>

Code	<pre>#include<stdio.h> void main () { printf ("%c %c",1,4,3,1,4,3,1,4,3,1,4,3,1,4,3,1,4,3,1,4,3,1,4,3,1,4,3,1,4,3,1,4,3,1); printf("\n%c\"If you are resisting something,you are feeding it.\"\\t\\t %c\",4,4); printf(\"\\n%c\\t\"Any energy you fight,you are feeding it.\"\\t\\t\\t %c\",3,3); printf(\"\\n%c\\t\\t\"If you are pushing something away,\"\\t\\t\\t %c\",1,1); printf(\"\\n%c\\t\\t\"You are inviting it to stay.\"by Michel singer.\\t %c\",4,4); printf(\"\\n%c %c\",1,4,3,1,4,3,1,4,3,1,4,3,1,4,3,1,4,3,1,4,3,1,4,3,1,4,3,1,4,3,1,4,3,1); }</pre>																		
Output Screenshot:																			
Question :	<p>1. Have you learnt about ASCII values for different symbols other than smile, diamond and heart? If yes, then mention any 5 ASCII symbols and their values in tabular format.</p> <p>Ans. Yes I learnt about ASCII values for different symbols here are few of them which I learnt:</p> <table><tr><th>Sr. No</th><th>Symbol</th><th>ASCII Values</th></tr><tr><td>1</td><td>@</td><td>64</td></tr><tr><td>2</td><td>?</td><td>63</td></tr><tr><td>3</td><td>^</td><td>94</td></tr><tr><td>4</td><td>;</td><td>59</td></tr><tr><td>5</td><td>.</td><td>149</td></tr></table>	Sr. No	Symbol	ASCII Values	1	@	64	2	?	63	3	^	94	4	;	59	5	.	149
Sr. No	Symbol	ASCII Values																	
1	@	64																	
2	?	63																	
3	^	94																	
4	;	59																	
5	.	149																	

Practical 1.2	<p>Write your bio-data using Escape Sequences. And you have to take your Basic Information as user input. It should contain the following content. It should contain the following content.</p> <p>Expected Outcome:</p> <p>Draw flowchart, write Algorithm and program for given scenario. Also attach screenshot of output.</p>
Algorithm	<p>Step 1: Start.</p> <p>Step 2: Print design of #= using printf.</p> <p>Step 3: Print BIO DATA with having centre alignment with help if \t escaped sequence.</p> <p>Step 4: Print your basic information, education qualification.</p> <p>Step 5: Use 149 as ASCII value for •.</p> <p>Step 6: Print other information and Thank you at end.</p> <p>Step 7: End.</p>
Flow chart	 <pre> graph TD Start([Start]) --> PrintDesign[/Print # = design using printf/] PrintDesign --> PrintInfo[/Print your basic information and education qualification/] PrintInfo --> UseASCII[Use 149 as ASCII Value for .] UseASCII --> PrintOther[/Print other information and thank you at end/] PrintOther --> End([End]) </pre>

Code	<pre> void main() { printf("#=====#=====#=====#=====#=====#=====#=====#=====#"); printf("\n BIO-DATA "); printf("\n#=====#=====#=====#=====#=====#=====#=====#=====#"); printf("\n Basic Information "); printf("\n _____ "); printf("\n Name : Trushang Patel "); printf("\n Address : vankal mota faliya "); printf("\n Mobile Number : 9712947385 "); printf("\n Gender : Male "); printf("\n Date of Birth : 28 feb 2005 "); printf("\n\n Education Qualification "); printf("\n _____ "); printf("\n Exam Name of school Passing year Percentage "); printf("\n SSC Shree sadguru r.m.m.vidyalay 2020 78.5%c ",37); printf("\n Exam Name of school Passing year Percentage "); printf("\n HSC R.n.naik high school 2022 82%c ",37); printf("\n "); printf("\n Other Information "); printf("\n _____ "); printf("\n Technical Skills : Java,C,C++ "); printf("\n Hobbies : Swimming, Table Tennis "); printf("\n#=====#=====#=====#=====#=====#=====#=====#=====#"); printf("\n THANK YOU "); printf("\n#=====#=====#=====#=====#=====#=====#=====#=====#"); } </pre>
-------------	--

Output Screenshot

```

#####
                        BIO-DATA
#####
                        Basic Information
                        _____
                        Name       : Trushang Patel
                        Address    : vankal mota faliya
                        Mobile Number : 9712947385
                        Gender     : Male
                        Date of Birth : 28 feb 2005

                        Education Qualification
                        _____
|Exam | Name of school | Passing year|Percentage |
|SSC  | Shree sadguru r.m.m.vidyalay | 2020      |78.5%   |
|Exam | Name of school | Passing year|Percentage |
|HSC  | R.n.naik high school | 2022      |82%     |

                        Other Information
                        _____
                        Technical Skills : Java,C,C++
                        Hobbies          : Swimming, Table Tennis
#####
                        THANK YOU
#####

```

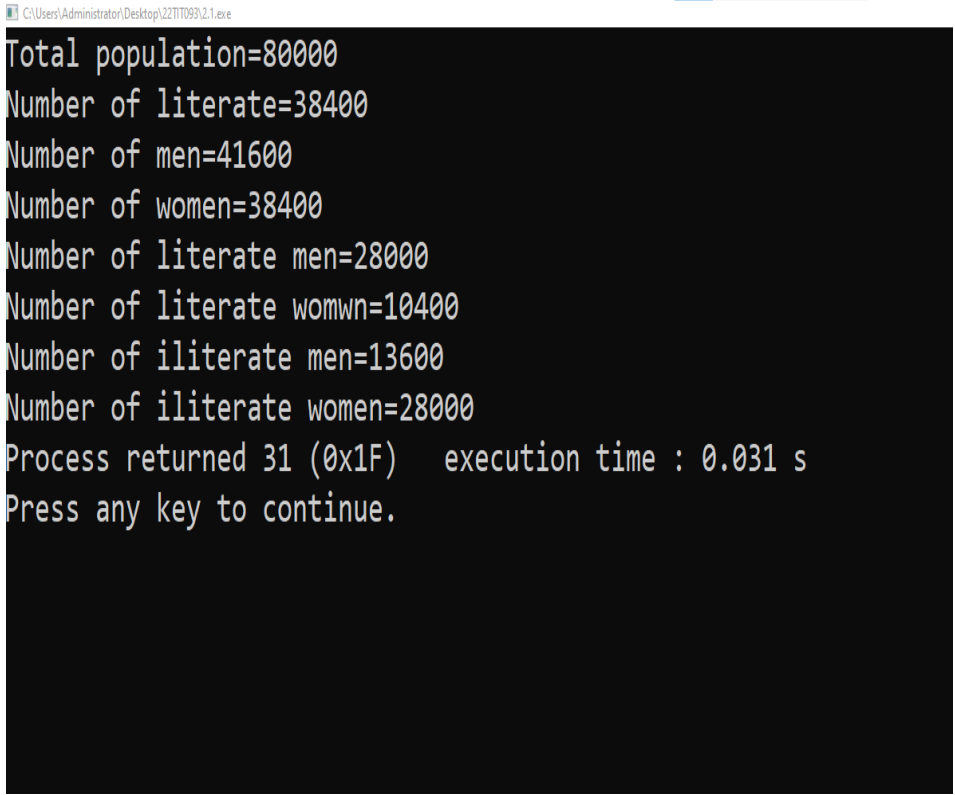
Question:

1. What is the purpose of using escape sequences? Answer in one or two statements. Mention any 5 escape sequences used regularly along with their purpose.

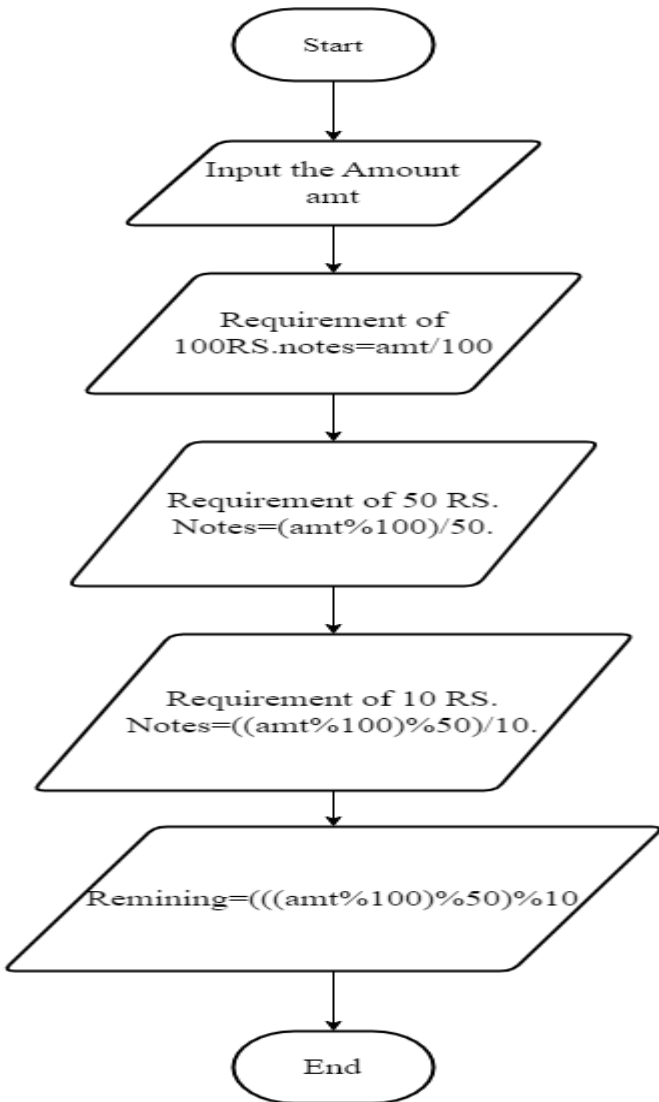
Sr. No	Escape Sequence	Purpose
1	\n	New line
2	\t	Horizontal tab
3	\\	Backlash
4	\a	Alarm or Beep
5	\"	Double Quote

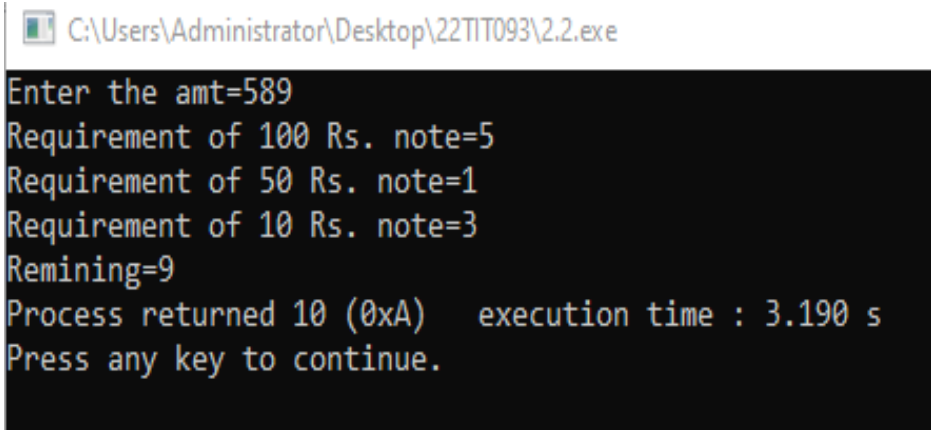
Sign:**Grade:**

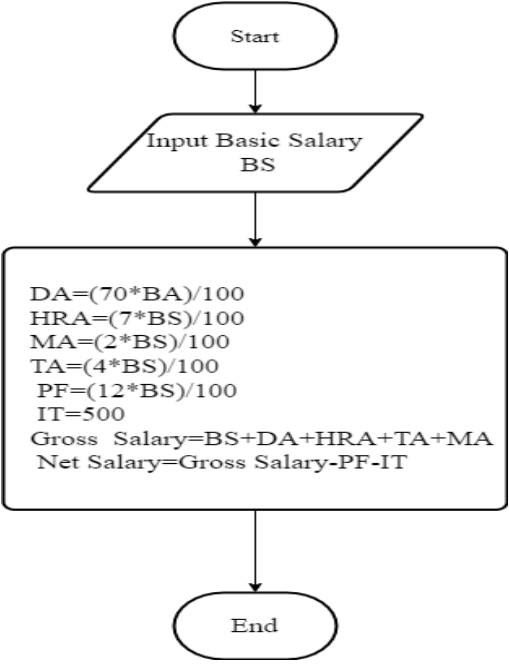
	SET-2: Constants, Variables & Data Types in 'C'
Practical 2.1	In a town, the percentage of men is 52. The percentage of total literacy is 48. If total percentage of literate men is 35 of the total population, write a program to find the total number of illiterate men and women if the population of the town is 80,000.
Algorithm	<p>Step 1: Start.</p> <p>Step 2: Population $p = 80000$.</p> <p>Step 3: Total men $m = (52 * p) / 100$.</p> <p>Step 4: Total women $w = p - m$</p> <p>Step 5: Total literacy $l = (48 * p) / 100$.</p> <p>Step 6: Total literate men $lm = (35 * p) / 100$.</p> <p>Step 7: Total literate women $lw = l - lm$.</p> <p>Step 8: Total illiterate men $ilm = m - lm$.</p> <p>Step 9: Total illiterate women $ilw = w - lw$.</p> <p>Step 10: End.</p>
Flow chart	<pre> graph TD Start([Start]) --> Input[/Input p=80000/] Input --> Process[m=(52*p)/100 w=p-m l=(48*p)/100 lm=(35*p)/100 lw=l-lm ilm=m-lm ilw=w-lw] Process --> End([End]) </pre>

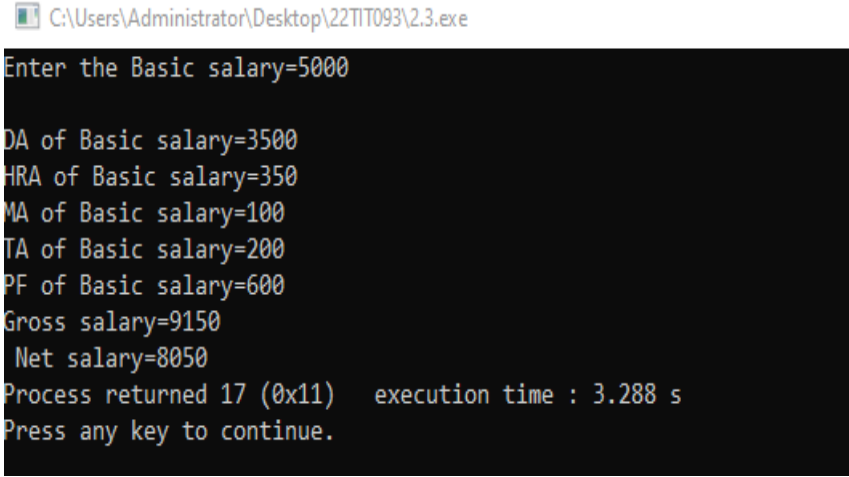
Code	<pre> void main() { int pop=80000,men,women,lit,litmen,litwomen,iliteratemen,iliteratewomen; printf("Total population=%d\n",pop); lit=pop*48/100; printf("Number of literate=%d\n",lit); men=pop*52/100; printf("Number of men=%d\n",men); women=pop-men; printf("Number of women=%d\n",women); litmen=pop*35/100; printf("Number of literate men=%d\n",litmen); litwomen=lit-litmen; printf("Number of literate womwn=%d\n",litwomen); iliteratemen=men-litmen; printf("Number of iliterate men=%d\n",iliteratemen); iliteratewomen=women-litwomen; printf("Number of iliterate women=%d",iliteratewomen); } </pre>
Output Screenshot	 <pre> C:\Users\Administrator\Desktop\22IT109\2.1.exe Total population=80000 Number of literate=38400 Number of men=41600 Number of women=38400 Number of literate men=28000 Number of literate womwn=10400 Number of iliterate men=13600 Number of iliterate women=28000 Process returned 31 (0x1F) execution time : 0.031 s Press any key to continue. </pre>

	<table><tr><td>Sr. No</td><td>Get outcome</td><td>Value</td></tr><tr><td>1</td><td>Total population</td><td>80000</td></tr><tr><td>2</td><td>Number of illiterates (Men+ Women)</td><td>38400</td></tr><tr><td>3</td><td>Number of men</td><td>41600</td></tr><tr><td>4</td><td>Number of literate women</td><td>28000</td></tr><tr><td>5</td><td>Number of illiterate men</td><td>13600</td></tr><tr><td>6</td><td>Number of women</td><td>38400</td></tr><tr><td>7</td><td>Number of literate women</td><td>10400</td></tr><tr><td>8</td><td>Number of illiterate women</td><td>28000</td></tr></table>	Sr. No	Get outcome	Value	1	Total population	80000	2	Number of illiterates (Men+ Women)	38400	3	Number of men	41600	4	Number of literate women	28000	5	Number of illiterate men	13600	6	Number of women	38400	7	Number of literate women	10400	8	Number of illiterate women	28000
Sr. No	Get outcome	Value																										
1	Total population	80000																										
2	Number of illiterates (Men+ Women)	38400																										
3	Number of men	41600																										
4	Number of literate women	28000																										
5	Number of illiterate men	13600																										
6	Number of women	38400																										
7	Number of literate women	10400																										
8	Number of illiterate women	28000																										
Question:	<p>1. Has this scenario helped you to learn about integer and float datatype? If yes, then mention the requirement of integer and float datatype.</p> <p>Ans: Integer data type is used when calculation is to be done of number having non decimal value while float data type is required when calculation is to be done of numbers having decimal value.</p>																											

Practical2.2	A Big bazaar cashier has currency notes of denominations 10,50 and 100. If the amount to be withdrawn is input through the keyboard in hundreds, find the total number of currency notes of each denomination the cashier will have to give to the withdrawer.
Algorithm	<p>Step 1: Start</p> <p>Step 2: Input the amount amt.</p> <p>Step 3: Requirement of 100 Rs. Notes = $\text{amt}/100$.</p> <p>Step 4: Requirement of 50 Rs. Notes = $(\text{amt}\%100)/50$.</p> <p>Step 5: Requirement of 10 Rs. Notes = $((\text{amt}\%100)\%50)/10$.</p> <p>Step 6: Remining= $((\text{amt}\%100)\%50)\%10$.</p> <p>Step 10: End.</p>
Flow chart	 <pre> graph TD Start([Start]) --> Input[/Input the Amount
amt/] Input --> Req100[/Requirement of
100RS.notes=amt/100/] Req100 --> Req50[/Requirement of 50 RS.
Notes=(amt%100)/50./] Req50 --> Req10[/Requirement of 10 RS.
Notes=((amt%100)%50)/10./] Req10 --> Remining[/Remining=((amt%100)%50)%10/] Remining --> End([End]) </pre>

Code	<pre>void main() { int amt; printf("Enter the amt="); scanf("%d",&amt); printf("Requirement of 100 Rs. note=%d\n",amt/100); printf("Requirement of 50 Rs. note=%d\n",(amt%100)/50); printf("Requirement of 10 Rs. note=%d\n",((amt%100)%50)/10); printf("Remining=%d",(((amt%100)%50)%10)); }</pre>																
Output Screenshot	<div></div> <table><tr><th>Sr. No</th><th>Note requirements</th><th>Count</th></tr><tr><td>1</td><td>Requirement of 100 Rs. Notes</td><td>5</td></tr><tr><td>2</td><td>Requirement of 100 Rs. Notes</td><td>1</td></tr><tr><td>3</td><td>Requirement of 100 Rs. Notes</td><td>3</td></tr></table>	Sr. No	Note requirements	Count	1	Requirement of 100 Rs. Notes	5	2	Requirement of 100 Rs. Notes	1	3	Requirement of 100 Rs. Notes	3				
Sr. No	Note requirements	Count															
1	Requirement of 100 Rs. Notes	5															
2	Requirement of 100 Rs. Notes	1															
3	Requirement of 100 Rs. Notes	3															
Question:	<div><div>1. Have you learned about how scanf function can be used to collect the user input? Give the correct answer for the following table:</div><table><tr><th>Sr. No.</th><th>Data Type</th><th>Format Specifier</th><th>Example of data</th></tr><tr><td>1</td><td>Integer</td><td>%d</td><td>7</td></tr><tr><td>2</td><td>Float</td><td>%f</td><td>7.5</td></tr><tr><td>3</td><td>char</td><td>%c</td><td>‘a’</td></tr></table></div>	Sr. No.	Data Type	Format Specifier	Example of data	1	Integer	%d	7	2	Float	%f	7.5	3	char	%c	‘a’
Sr. No.	Data Type	Format Specifier	Example of data														
1	Integer	%d	7														
2	Float	%f	7.5														
3	char	%c	‘a’														

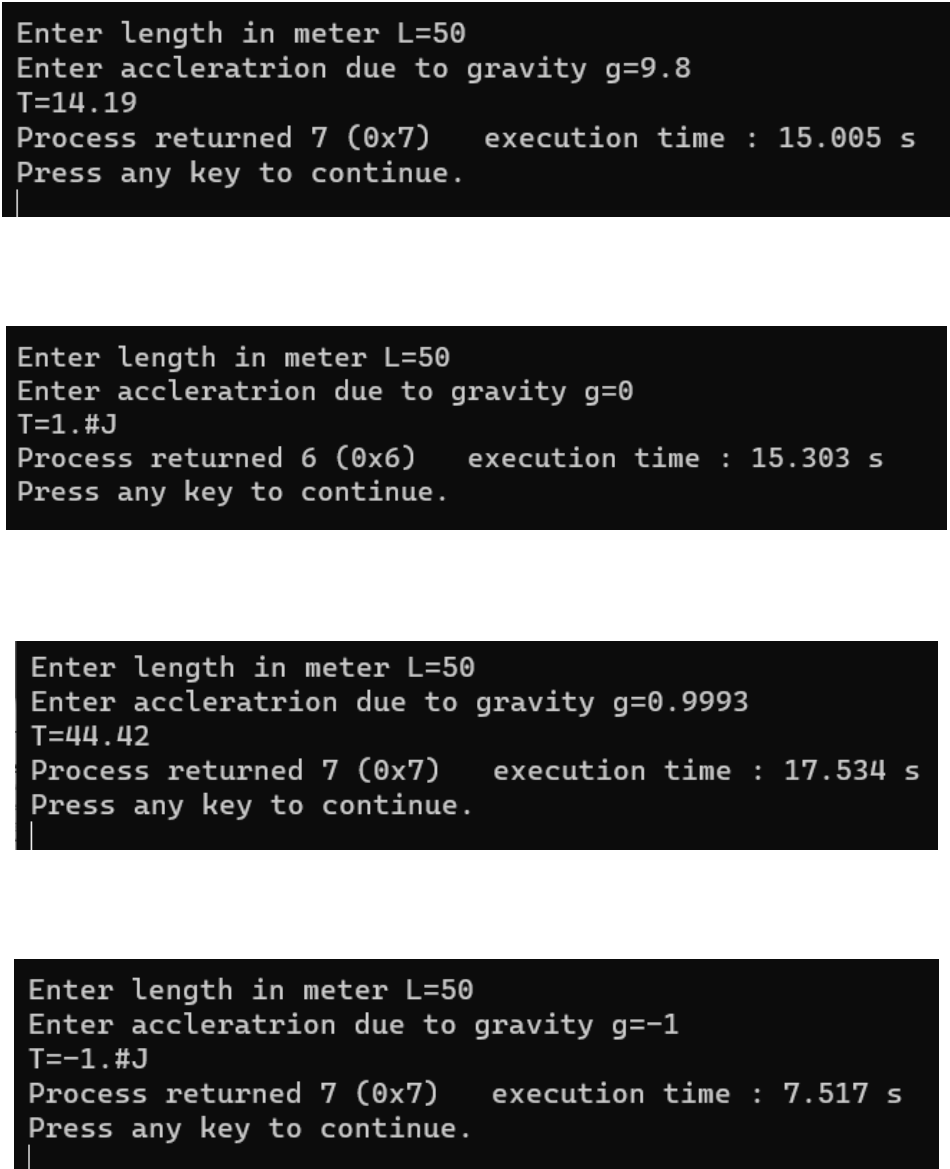
Practical 2.3	<p>Write a program to calculate Net Salary. User has to input Basic Salary and Output should be: Enter Basic Salary: 5000 (e.g., 5000) Allowances: DA = 70% of Basic Salary HRA = 7% of Basic Salary MA = 2% of Basic Salary TA = 4% of Basic Salary Deduction: PF = 12% of Basic Salary IT = any value (e.g., 500)</p> <hr/> <p>Gross Salary = Basic Salary + Allowances Net Salary = Gross Salary – Deduction</p>
Algorithm	<p>Step 1: Start. Step 2: Input basic salary bs. Step 3: $DA = (70 * bs) / 100$. $HRA = (7 * bs) / 100$. $MA = (2 * bs) / 100$. $TA = (4 * bs) / 100$. $PF = (12 * bs) / 100$. $IT = 500$. Gross salary = BS+DA+HRA+MA+TA. Net salary = Gross Salary-PF-IT. Step 4: End.</p>
Flow chart	 <pre> graph TD Start([Start]) --> Input[/Input Basic Salary BS/] Input --> Process[DA=(70*BS)/100 HRA=(7*BS)/100 MA=(2*BS)/100 TA=(4*BS)/100 PF=(12*BS)/100 IT=500 Gross Salary=BS+DA+HRA+TA+MA Net Salary=Gross Salary-PF-IT] Process --> End([End]) </pre>

Code	<pre> void main() { int BS,IT=500,DA,HRA,MA,TA,PF,GS,NS,Allowances,Deduction; printf("Enter the Basic salary="); scanf("%d",&BS); DA=(BS*70)/100; printf("\nDA of Basic salary=%d",DA); HRA=(BS*7)/100; printf("\nHRA of Basic salary=%d",HRA); MA=(BS*2)/100; printf("\nMA of Basic salary=%d",MA); TA=(BS*4)/100; printf("\nTA of Basic salary=%d",TA); Allowances=DA+HRA+MA+TA; PF=(BS*12)/100; printf("\nPF of Basic salary=%d",PF); GS=BS+Allowances; printf("\nGross salary=%d",GS); Deduction=PF+IT; NS=GS-Deduction; printf("\n Net salary=%d",NS); } </pre>
Output Screenshot	 <pre> C:\Users\Administrator\Desktop\22TIT093\2.3.exe Enter the Basic salary=5000 DA of Basic salary=3500 HRA of Basic salary=350 MA of Basic salary=100 TA of Basic salary=200 PF of Basic salary=600 Gross salary=9150 Net salary=8050 Process returned 17 (0x11) execution time : 3.288 s Press any key to continue. </pre>

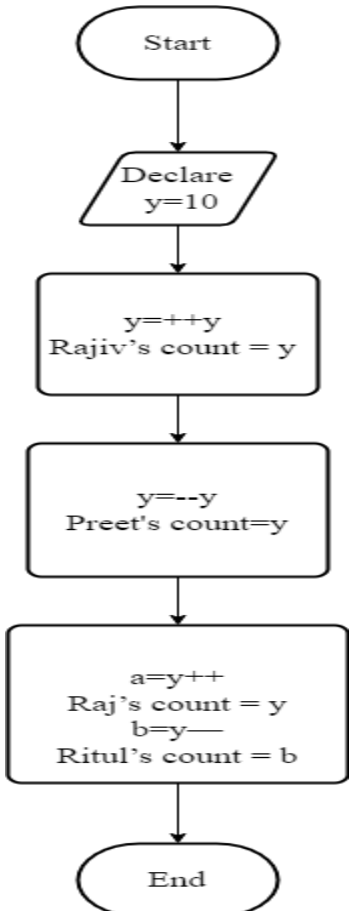
	<table><tr><th>Sr.No.</th><th>Input/Outputs</th><th>Amount</th></tr><tr><td>1</td><td>Basic salary</td><td>5000</td></tr><tr><td>2</td><td>DA of Basic salary</td><td>3500</td></tr><tr><td>3</td><td>HRA of Basic salary</td><td>350</td></tr><tr><td>4</td><td>MA of Basic salary</td><td>100</td></tr><tr><td>5</td><td>TA of Basic salary</td><td>200</td></tr><tr><td>6</td><td>PF of Basic salary</td><td>600</td></tr><tr><td>7</td><td>Gross Salary</td><td>9150</td></tr><tr><td>8</td><td>Net Salary</td><td>8050</td></tr></table>	Sr.No.	Input/Outputs	Amount	1	Basic salary	5000	2	DA of Basic salary	3500	3	HRA of Basic salary	350	4	MA of Basic salary	100	5	TA of Basic salary	200	6	PF of Basic salary	600	7	Gross Salary	9150	8	Net Salary	8050
Sr.No.	Input/Outputs	Amount																										
1	Basic salary	5000																										
2	DA of Basic salary	3500																										
3	HRA of Basic salary	350																										
4	MA of Basic salary	100																										
5	TA of Basic salary	200																										
6	PF of Basic salary	600																										
7	Gross Salary	9150																										
8	Net Salary	8050																										
Question:	<p>1. Have you learned about various data types that can be suitably used for this problem? Do mention which data types can be used and why? Also mention the difference between the outputs.</p> <p>Ans: - The basic data types used in C are as follows:</p> <ul style="list-style-type: none">• char: The most basic data type in C. It stores a single character and requires a single byte of memory in almost all compilers.• int: As the name suggests, an int variable is used to store an integer.• float: It is used to store decimal numbers (numbers with floating point value) with single precision.• double: It is used to store decimal numbers (numbers with floating point value) with double precision.																											

Sign:**Grade:**

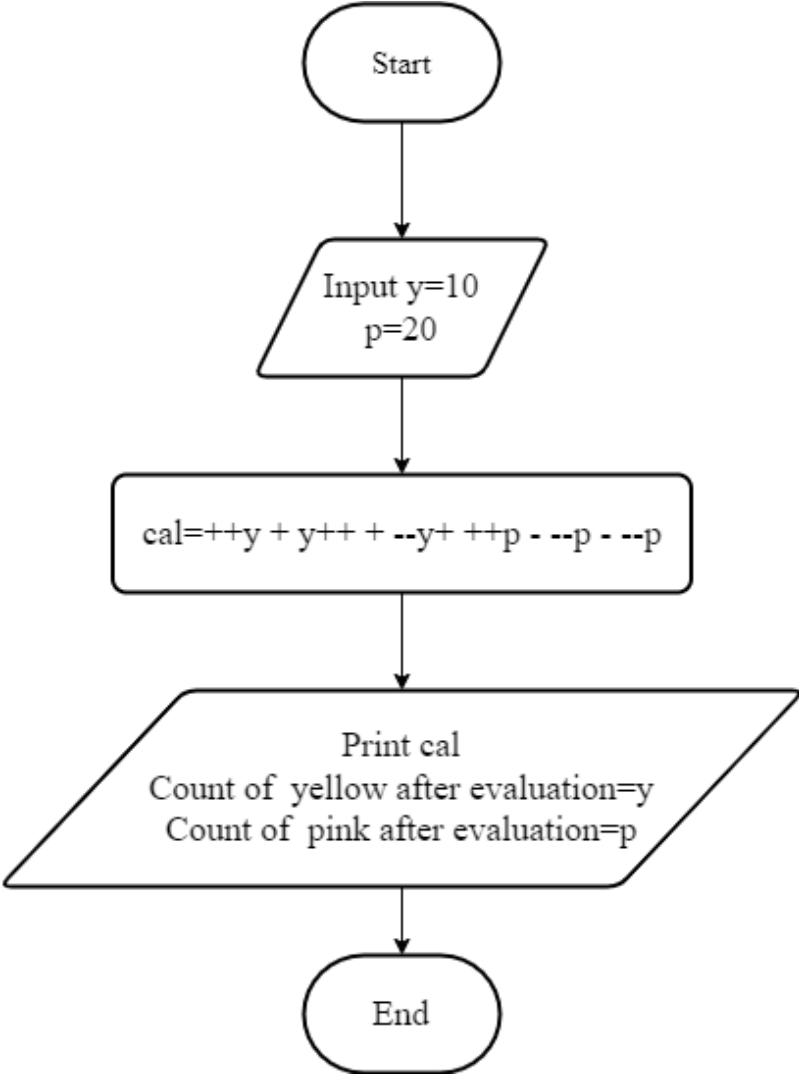
	SET-3 : Operators and Expression in 'C'
Practical3.1	<p>Write a program that takes the length of the pendulum as input and then calculate the time period of the pendulum. Provided that, $T=2\pi\sqrt{L/G}$. Define the value of π as 3.14 and take L as the length of the pendulum and G as the acceleration of gravity either in m/s or as input from the keyboard. Display the time period rounded to 2 decimal places. Hint:</p> <p>Use Math.h header file, use #define for specifying the value of π.</p>
Algorithm	<p>Step 1: Start</p> <p>Step 2: Define pi = 3.14.</p> <p>Step 3: Input L in metre and g in m/s</p> <p>Step 4: $T = 2*\pi*\text{sqrt}(L/g)$</p> <p>Step 5: Print T.</p> <p>Step 6: End</p>
Flow chart	<pre> graph TD Start([Start]) --> Input[/Define pi=3.14 Input L in meter and G is m/s2/] Input --> Process[T=2*pi*sqrt(L/G)] Process --> Output[/Print T/] Output --> End([End]) </pre>

Code	<pre>#include<stdio.h> #include<conio.h> #include<math.h> #define pi 3.14 void main() { float T,L,g; printf("Enter length in meter L="); scanf("%f",&L); printf("Enter accleratrion due to gravity g="); scanf("%f",&g); T=2*pi*sqrt(L/g); printf("T=%f",T); }</pre>
Output Screenshot	 <p>The first screenshot shows the program running with L=50 and g=9.8, resulting in T=14.19. The second screenshot shows L=50 and g=0, resulting in T=1. The third screenshot shows L=50 and g=0.9993, resulting in T=44.42. The fourth screenshot shows L=50 and g=-1, resulting in T=-1. Each screenshot also displays the process return code, execution time, and a prompt to press any key to continue.</p> <pre>Enter length in meter L=50 Enter accleratrion due to gravity g=9.8 T=14.19 Process returned 7 (0x7) execution time : 15.005 s Press any key to continue. Enter length in meter L=50 Enter accleratrion due to gravity g=0 T=1.#J Process returned 6 (0x6) execution time : 15.303 s Press any key to continue. Enter length in meter L=50 Enter accleratrion due to gravity g=0.9993 T=44.42 Process returned 7 (0x7) execution time : 17.534 s Press any key to continue. Enter length in meter L=50 Enter accleratrion due to gravity g=-1 T=-1.#J Process returned 7 (0x7) execution time : 7.517 s Press any key to continue. </pre>

	<table><tr><th rowspan="2">Sr. No.</th><th colspan="2">Input</th><th>Output</th></tr><tr><th>Length</th><th>Gravity</th><th>Time Calculated(seconds)</th></tr><tr><td>1.</td><td>50 m</td><td>9.8 m/s²</td><td>14.19</td></tr><tr><td>2.</td><td>50 m</td><td>0 m/s²</td><td>Error</td></tr><tr><td>3.</td><td>50 m</td><td>0.9993 g</td><td>44.42</td></tr><tr><td>4.</td><td>50 m</td><td>-1 g</td><td>Error</td></tr></table>	Sr. No.	Input		Output	Length	Gravity	Time Calculated(seconds)	1.	50 m	9.8 m/s ²	14.19	2.	50 m	0 m/s ²	Error	3.	50 m	0.9993 g	44.42	4.	50 m	-1 g	Error
Sr. No.	Input		Output																					
	Length	Gravity	Time Calculated(seconds)																					
1.	50 m	9.8 m/s ²	14.19																					
2.	50 m	0 m/s ²	Error																					
3.	50 m	0.9993 g	44.42																					
4.	50 m	-1 g	Error																					
Question:	<p>1. Have you learned about, how math function is useful for calculating square root? Which datatype is supported by all math functions? Also mention any 5 math functions with their purpose.</p> <p>Ans. Float datatype is supported by all math functions.</p> <table><tr><th>Sr. No</th><th>Math Function</th><th>Description</th></tr><tr><td>1</td><td>pow</td><td>Returns the power of given number.</td></tr><tr><td>2</td><td>abs</td><td>Return the absolute value of given number.</td></tr><tr><td>3</td><td>ceil</td><td>Returns the integer value which is greater than or equal to given number.</td></tr><tr><td>4</td><td>floor</td><td>Returns the integer value which is less than or equal to given number.</td></tr><tr><td>5</td><td>log</td><td>Returns the natural logarithm (Base-e logarithm) of given number.</td></tr></table>	Sr. No	Math Function	Description	1	pow	Returns the power of given number.	2	abs	Return the absolute value of given number.	3	ceil	Returns the integer value which is greater than or equal to given number.	4	floor	Returns the integer value which is less than or equal to given number.	5	log	Returns the natural logarithm (Base-e logarithm) of given number.					
Sr. No	Math Function	Description																						
1	pow	Returns the power of given number.																						
2	abs	Return the absolute value of given number.																						
3	ceil	Returns the integer value which is greater than or equal to given number.																						
4	floor	Returns the integer value which is less than or equal to given number.																						
5	log	Returns the natural logarithm (Base-e logarithm) of given number.																						

Practical3.2a)	<p>Let us understand the working of Pre-increment, Post-increment, Pre-decrement and Post-decrement</p> <p>a) Consider a scenario where, Boys are playing in the park and collecting and removing the yellow balls in/from the bucket based on teacher's instruction. Let's say there are already 10 Yellow balls present in a bucket. Following is the sequence of the instructions given by the teacher for adding/removing the balls.</p> <p>i. Rajiv: ++ Yellow ii. Preet: --Yellow iii. Raj: Yellow++ iv. Ritul: Yellow- -</p>
Algorithm	<p>Step 1: Start.</p> <p>Step 2: $y = 10$.</p> <p>Step 3: $y = ++y$ and print Rajiv's count = y.</p> <p>Step 4: $y = --y$ and print Preet's count = y.</p> <p>Step 5: $a = ++y$ and print Raj's count = a.</p> <p>Step 6: $y = y--$ and print Ritul's count = y.</p> <p>Step 7: End.</p>
Flow chart	 <pre> graph TD Start([Start]) --> Declare[/Declare y=10/] Declare --> Rajiv[y=++y Rajiv's count = y] Rajiv --> Preet[y=--y Preet's count=y] Preet --> RajRitul[a=y++ Raj's count = y b=y-- Ritul's count = b] RajRitul --> End([End]) </pre>

Code	<pre>#include<stdio.h> void main() { int y=10,a,b; printf("\n Count before execution: %d",y); y = ++y; printf("\n Rajiv's Count: %d",y); y = --y; printf("\n Preet's Count: %d",y); a = y++; printf("\n Raj's Count: %d",a); b = y--; printf("\n Ritul's Count: %d",b); printf("\n Count after execution: %d",b); }</pre>									
Output Screenshot	<div><pre>Count before execution: 10 Rajiv's Count: 11 Preet's Count: 10 Raj's Count: 10 Ritul's Count: 11 Count after execution: 11 Process returned 27 (0x1B) execution time : 3.405 s Press any key to continue.</pre></div> <table><tr><th>Sr. No</th><th>Instructions</th><th>Yellow</th></tr><tr><td>1</td><td>Count before execution</td><td>10</td></tr><tr><td>2</td><td>Count after execution</td><td>11</td></tr></table>	Sr. No	Instructions	Yellow	1	Count before execution	10	2	Count after execution	11
Sr. No	Instructions	Yellow								
1	Count before execution	10								
2	Count after execution	11								

Practical3.2b)	b) Consider another scenario where boys and girls both are asked to add/remove yellow and pink balls from the bucket respectively. Currently there are 10 Yellow balls in the bucket and 20 Pink balls. Teacher has given the sequence of instructions as below for adding/removing the balls. Calculate = ++Yellow + Yellow++ + --Yellow + ++Pink - --Pink - --Pink. Get the count of yellow and pink balls after evaluating above given scenario.
Algorithm	Step 1: Start. Step 2: Input y = 10, p = 20. Step 3: cal=++y + y++ + --y + ++p - --p- --p. Step 4: Count of Yellow after evaluation =y. Count of pink after evaluation=p. Step 5: End.
Flow chart	 <pre> graph TD Start([Start]) --> Input[/Input y=10 p=20/] Input --> Process[cal=++y + y++ + --y + ++p - --p - --p] Process --> Output[/Print cal Count of yellow after evaluation=y Count of pink after evaluation=p/] Output --> End([End]) </pre>

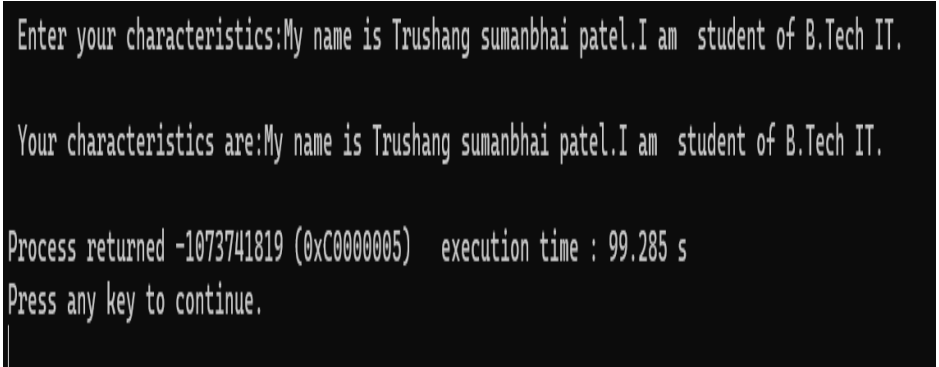
Code	<pre>#include<stdio.h> void main() { int y,p,cal; printf("\n Enter the value of y:"); scanf("%d",&y); printf("\n Enter the value of p:"); scanf("%d",&p); cal = (++y) + (y++) + (--y) + (++p) - (--p) - (--p); printf("\n Value of calculate is: %d",cal); printf("\n Count of yellow after evaluation: %d",y); printf("\n Count of pink after evaluation: %d",p); }</pre>												
Output Screenshot	<div><pre>Enter the value of y:10 Enter the value of p:20. Value of calculate is: 15 Count of yellow after evaluation: 11 Count of pink after evaluation: 19 Process returned 36 (0x24) execution time : 2.477 s Press any key to continue.</pre></div> <table><tr><th>Sr. No</th><th>Instruction</th><th>Yellow</th><th>Pink</th></tr><tr><td>1</td><td>Count before execution</td><td>10</td><td>20</td></tr><tr><td>2</td><td>Count after execution</td><td>11</td><td>19</td></tr></table>	Sr. No	Instruction	Yellow	Pink	1	Count before execution	10	20	2	Count after execution	11	19
Sr. No	Instruction	Yellow	Pink										
1	Count before execution	10	20										
2	Count after execution	11	19										
Question:	<p>1. Have you understood the working of Pre-increment, Post-increment, Pre-decrement and Post-decrement?</p> <p>Ans: - Yes through this practical I understood the working of Pre-increment, Post increment, Pre-decrement and Post-decrement.</p>												

Practical 3.3	Write a C program to swap two numbers (use two variables for collecting value from user) without using third variable. (Hint: Use arithmetic operators).
Algorithm	<p>Step 1: Start.</p> <p>Step 2: Input the value of a, b.</p> <p>Step 3: Value of a before swapping = a. Value of b before swapping = b.</p> <p>Step 4: $a = a + b$</p> <p>Step 5: $b = a - b$</p> <p>Step 6: $a = a - b$</p> <p>Step 7: Value of a after swapping = a. Value of b after swapping = b.</p> <p>Step 8: End</p>
Flow chart	<pre> graph TD Start([Start]) --> Input[/Input value of a&b/] Input --> Process[a = a + b b = a - b a = a - b] Process --> Output[/Print value of new a&b/] Output --> End([End]) </pre>

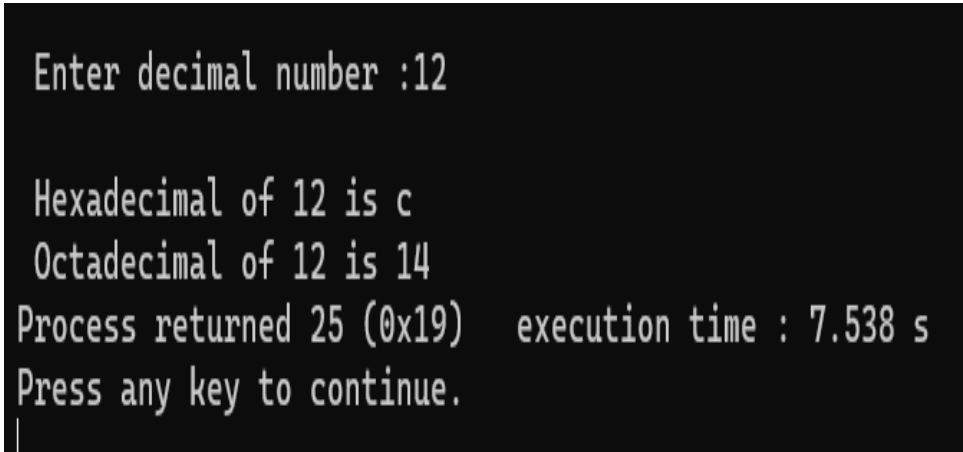
Code	<pre>void main() { int a,b; printf("Enter the value of a="); scanf("%d",&a); printf("Enter the value of b="); scanf("%d",&b); printf("Before Swapping"); printf("\na=%d\nb=%d",a,b); a=a+b; b=a-b; a=a-b; printf("\nAfter Swapping"); printf("\na=%d\nb=%d",a,b); }</pre>												
Output Screenshot	<div><pre>Enter the value of a=10 Enter the value of b=20 Before Swapping a=10 b=20 After Swapping a=20 b=10 Process returned 10 (0xA) execution time : 6.450 s Press any key to continue. </pre></div> <table><tr><th>Sr. No</th><th>Instruction</th><th>a</th><th>b</th></tr><tr><td>1</td><td>Before swapping</td><td>10</td><td>20</td></tr><tr><td>2</td><td>After swapping</td><td>20</td><td>10</td></tr></table>	Sr. No	Instruction	a	b	1	Before swapping	10	20	2	After swapping	20	10
Sr. No	Instruction	a	b										
1	Before swapping	10	20										
2	After swapping	20	10										
Question:	<p>1. Have you learned about, how we can use arithmetic operators for swapping the numbers?</p> <p>Ans. Yes, we can use arithmetic operators such as addition, subtraction for swapping of two numbers.</p>												

Sign:**Grade:**

	SET-4: Managing Input & Output Operations
Practical 4.1	a). Write something about your characteristics not more than 50 words using gets function and print out the same using puts function.
Algorithm	Step 1: Start Step 2: Describe Data [10] Step 3: Enter characteristics Step 4: Print characteristics Step 5: End
Flow chart	<pre>graph TD; Start([Start]) --> Declare[Declare Data [10]]; Declare --> Enter[/Enter characteristics/]; Enter --> Print[/Print characteristics/]; Print --> End([End]);</pre>

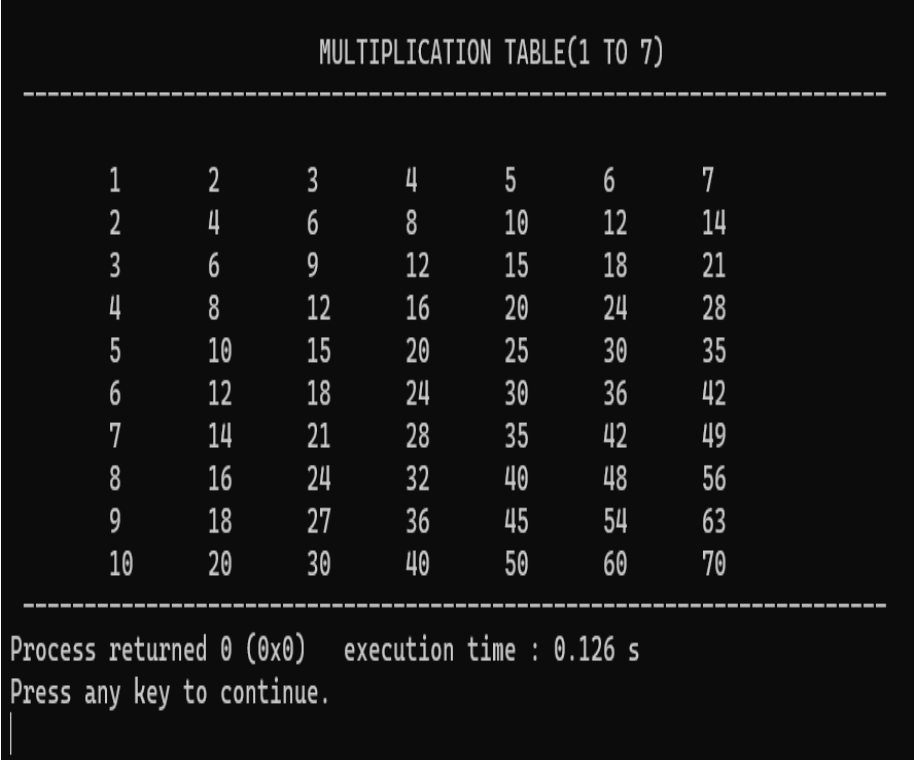
Code	<pre>#include<stdio.h> void main() { char data[10]; printf("\n Enter your characteristics:"); gets(data); printf("\n Your characteristics are:"); puts(data); }</pre>
Output Screenshot	 <p>The screenshot shows a terminal window with the following text:</p> <pre>Enter your characteristics:My name is Trushang sumanbhai patel.I am student of B.Tech IT. Your characteristics are:My name is Trushang sumanbhai patel.I am student of B.Tech IT. Process returned -1073741819 (0xC0000005) execution time : 99.285 s Press any key to continue. </pre>
Questions:	<p>1. What is the significance of using gets and puts? Are they acting as replacement of any function? How?</p> <p>Ans: The use of gets and puts is to print the function in c and to ask value from user in c respectively. Here they are acting as replacement of function printf and scanf in c programming. Where printf is used at that place the puts function is used to print the line in c and similarly to ask the value instead of scanf the function gets is used.</p>

Practical 4.1	<p>b) Write a program to convert the decimal number into octal and hexadecimal format. Print hexadecimal and octal values for given inputs in expected outcomes. (Hint: Use %o and %x)</p> <p>Expected outcome must be filled in below table:</p> <table><tr><th>Sr. No.</th><th>Inputs</th><th>Octal</th><th>Hexadecimal</th></tr><tr><td>1.</td><td>Your Roll No</td><td></td><td></td></tr><tr><td>2.</td><td>143</td><td></td><td></td></tr><tr><td>3.</td><td>0</td><td></td><td></td></tr><tr><td>4.</td><td>1</td><td></td><td></td></tr><tr><td>5.</td><td>-1</td><td></td><td></td></tr></table>	Sr. No.	Inputs	Octal	Hexadecimal	1.	Your Roll No			2.	143			3.	0			4.	1			5.	-1		
Sr. No.	Inputs	Octal	Hexadecimal																						
1.	Your Roll No																								
2.	143																								
3.	0																								
4.	1																								
5.	-1																								
Algorithm	<p>Step 1: Start</p> <p>Step 2: Describe no;</p> <p>Step 3: Print decimal value;</p> <p>Step 4: Hexadecimal form is %x;</p> <p>Step 5: Octal form is %o;</p> <p>Step 6: Print Hexadecimal form and octal form.</p> <p>Step 7: Stop</p>																								
Flow chart	<pre>graph TD; Start([Start]) --> Declare[Declare no]; Declare --> Process[Hexadecimal form is %x Octal decimal form is %o]; Process --> Output[/Print Hexadecimal form and octal decimal form/]; Output --> End([End]);</pre>																								

Code	<pre>#include<stdio.h> void main() { int no; printf("\n Enter decimal number :"); scanf("%d",&no); printf("\n Hexadecimal of %d is %x",no,no); printf("\n Octadecimal of %d is %o",no,no); }</pre>																								
Output Screenshot	<div></div> <table><tr><th>Sr. No</th><th>Inputs</th><th>Octal</th><th>Hexadecimal</th></tr><tr><td>1</td><td>116</td><td>164</td><td>74</td></tr><tr><td>2</td><td>143</td><td>217</td><td>8f</td></tr><tr><td>3</td><td>0</td><td>0</td><td>0</td></tr><tr><td>4</td><td>1</td><td>1</td><td>1</td></tr><tr><td>5</td><td>-1</td><td>37777777</td><td>ffffff</td></tr></table>	Sr. No	Inputs	Octal	Hexadecimal	1	116	164	74	2	143	217	8f	3	0	0	0	4	1	1	1	5	-1	37777777	ffffff
Sr. No	Inputs	Octal	Hexadecimal																						
1	116	164	74																						
2	143	217	8f																						
3	0	0	0																						
4	1	1	1																						
5	-1	37777777	ffffff																						

Practical 4.2	Write a C Program to Print multiplication table from 1 to 7 to achieve the following output. (Use #define directives and do while loop).
Algorithm	<p>Step 1: Start</p> <p>Step 2: Declare i</p> <p>Step 3: Using loop print number for (i=1;i<=10;i++)</p> <p>Step 4: Use another loop for column such as (j=1;j<=7;j++)</p> <p>Step 5: Print the number and multiply it as i*j.</p> <p>Step 6: Stop.</p>
Flow chart	<pre> graph TD Start([Start]) --> Declare[/Declare i/] Declare --> Loop1{for i=1 to 10} Loop1 -- Yes --> Loop2{for j=1 to 7} Loop2 -- Yes --> Print[/Print nuumber i*j/] Print --> Loop1 Loop2 -- No --> End([End]) Loop1 -- No --> End </pre>

\

Code	<pre> #include<stdio.h> main() { int i,j; printf("\n \t \t \t MULTIPLICATION TABLE(1 TO 7)"); printf("\n ----- -----"); printf("\n"); for(i = 1; i<=10; i = i+1) { printf("\n"); for(j=1; j<=7; j++) { printf("\t%d",i*j); } } printf("\n ----- -----"); } </pre>
Output Screenshot	 <pre> MULTIPLICATION TABLE(1 TO 7) ----- 1 2 3 4 5 6 7 2 4 6 8 10 12 14 3 6 9 12 15 18 21 4 8 12 16 20 24 28 5 10 15 20 25 30 35 6 12 18 24 30 36 42 7 14 21 28 35 42 49 8 16 24 32 40 48 56 9 18 27 36 45 54 63 10 20 30 40 50 60 70 ----- Process returned 0 (0x0) execution time : 0.126 s Press any key to continue. </pre>

Sign:**Grade:**

SET-5: Conditional Statements & Branching																																			
Practical5.1	<p>Write a C program for the given scenario from the flowchart. Note that you have to enter your own height in centimetres.</p> <p>Expected outcome must be like this:</p> <table><tr><td>Sr. No.</td><td>Inputs (cm)</td><td>Dwarf</td><td>Average</td><td>Tall</td><td>Abnormal</td></tr><tr><td>1.</td><td>Your Height</td><td></td><td></td><td></td><td></td></tr><tr><td>2.</td><td>Your Mother's height</td><td></td><td></td><td></td><td></td></tr><tr><td>3.</td><td>Your Father's height</td><td></td><td></td><td></td><td></td></tr><tr><td>4.</td><td>Your Sibling's height</td><td></td><td></td><td></td><td></td></tr></table>					Sr. No.	Inputs (cm)	Dwarf	Average	Tall	Abnormal	1.	Your Height					2.	Your Mother's height					3.	Your Father's height					4.	Your Sibling's height				
Sr. No.	Inputs (cm)	Dwarf	Average	Tall	Abnormal																														
1.	Your Height																																		
2.	Your Mother's height																																		
3.	Your Father's height																																		
4.	Your Sibling's height																																		
Algorithm	<p>Step 1: Start</p> <p>Step 2: Declare H.</p> <p>Step 3: input H</p> <p>Step 4: check if $H < 150$</p> <p>Step 5: print Height is dwarf</p> <p>Step 6: check if $H \geq 150 \ \&\& \ H < 165$</p> <p>Step 7: print Height is Average.</p> <p>Step 8: Check if $H \geq 165 \ \&\& \ H \leq 195$</p> <p>Step 9: print Height is Tall.</p> <p>Step 10: If $H > 195$</p> <p>Step11: Print U have abnormal height.</p> <p>Step 12: Stop.</p>																																		
Flow chart	<pre>graph TD Start([Start]) --> DeclareH[/Declare H/] DeclareH --> H150{H < 150} H150 -- Yes --> Dwarf[Dwarf] H150 -- No --> H150_165{H >= 150 && H < 165} H150_165 -- Yes --> Avg[Avg height] H150_165 -- No --> H165_195{H >= 165 && H <= 195} H165_195 -- Yes --> Abnormal[Ab normal high] H165_195 -- No --> Tall[Tall] Dwarf --> End([End]) Avg --> End Abnormal --> End Tall --> End</pre>																																		

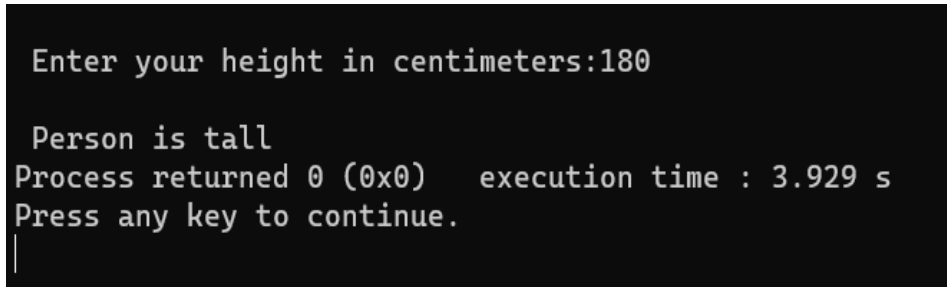
Code

```
#include<stdio.h>
main()

{
    int h;

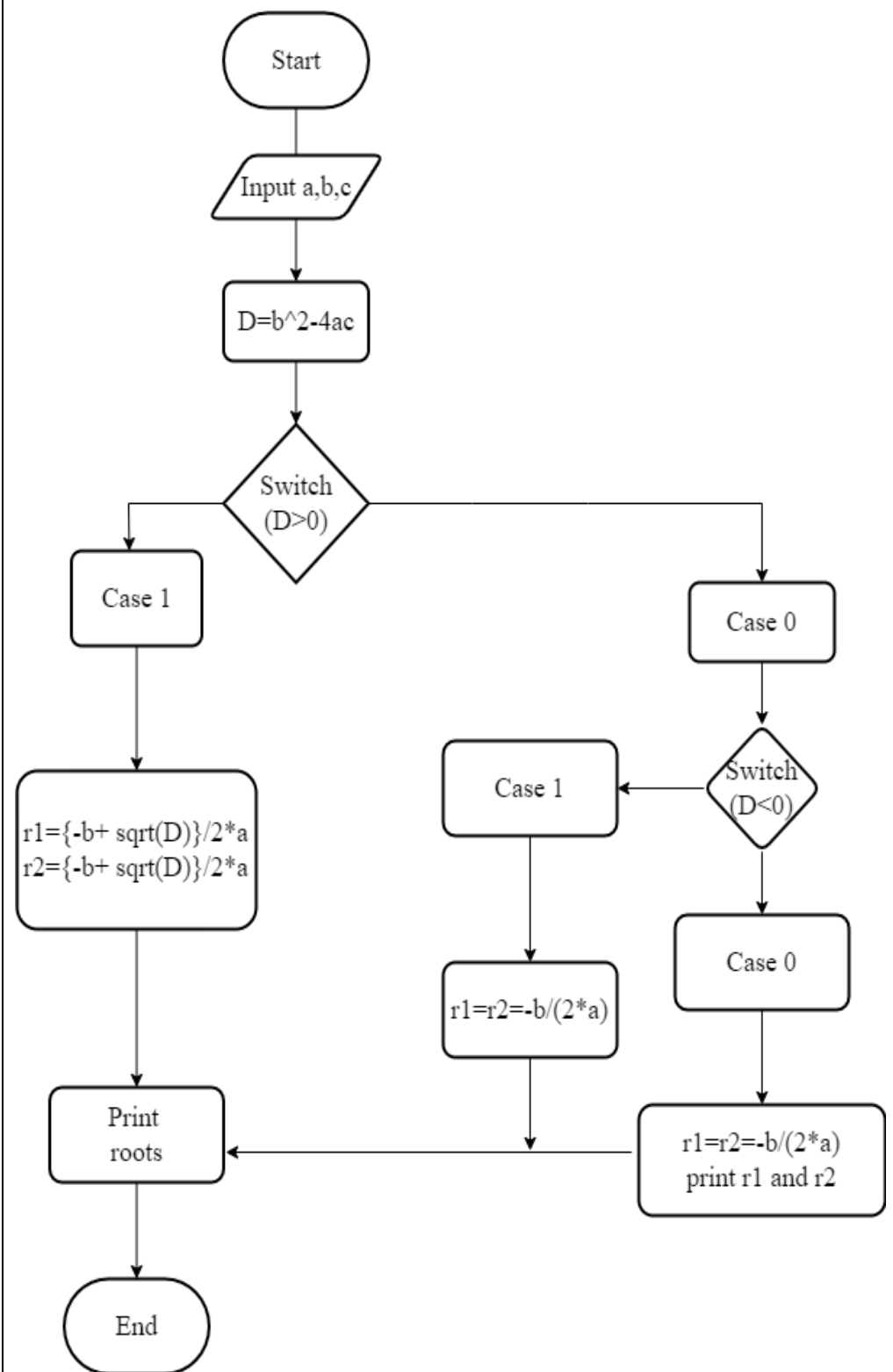
    printf("\n Enter your height in centimeters:");
    scanf("%d",&h);
    if(h<150)
    {
        printf("\n Person is dwarf");
    }
    else if(h>=150 && h<165)
    {
        printf("\n Person has average height");
    }
    else if(h>=165 && h<=195)
    {
        printf("\n Person is tall");
    }
    else
    {
        printf("\nPerson has abnormal height");
    }
}
```

Output Screenshot



Sr. No	Inputs(cm)	Dwarf	Average	Tall	Abnormal
1	Your hight			√	
2	Your mother's hight		√		
3	Your father's height			√	
4	Your sibling height	√			

Practical 5.2	<p>Write a C program to find all roots of a Quadratic equation using nested switch case. Take three user inputs from keyboard for finding the discriminant ($b^2 - 4ac$). Use the concept of nested switch case for finding the roots of equation. Get the outputs for roots till 2 decimal points only. Hint: Discriminant > 0 root1 = $(-b + \sqrt{\text{discriminant}}) / (2*a)$ root2 = $(-b - \sqrt{\text{discriminant}}) / (2*a)$ Discriminant < 0 root1 = root2 = $-b / (2*a)$ imaginary = $\sqrt{-\text{discriminant}} / (2*a)$ (eg. Print it as: i20.3, i.e. i followed by value) Discriminant = 0 root1 = root2 = $-b / (2*a)$ Expected outcome must be filled in below table:</p> <table><tr><th rowspan="2">Sr. No.</th><th colspan="3">Inputs</th><th rowspan="2">Root1</th><th rowspan="2">Root2</th><th rowspan="2">Imaginary</th></tr><tr><th>a</th><th>b</th><th>c</th></tr><tr><td>1.</td><td>1</td><td>2</td><td>3</td><td></td><td></td><td></td></tr><tr><td>2.</td><td>3</td><td>-7</td><td>-5</td><td></td><td></td><td></td></tr><tr><td>3.</td><td>9</td><td>12</td><td>4</td><td></td><td></td><td></td></tr></table>	Sr. No.	Inputs			Root1	Root2	Imaginary	a	b	c	1.	1	2	3				2.	3	-7	-5				3.	9	12	4			
Sr. No.	Inputs			Root1	Root2				Imaginary																							
	a	b	c																													
1.	1	2	3																													
2.	3	-7	-5																													
3.	9	12	4																													
Algorithm	<p>Step 1: Start Step 2: Enter value of a, b, c, D, r1, r2, n, img Step 3: switch D>0 if Case 0 { Switch (D<0) If case0 Print $r1=r2=(-b)/(2*a)$ Case1 Print $r1=r2=(-b)/(2*a)$ } Case 1 Print $r1=(-b+\sqrt{D})/(2*a);$ Print $r2= (-b-\sqrt{D})/(2*a);$ Step 4: Stop.</p>																															

**Flow
chart**

Code	<pre> #include<stdio.h> #include<math.h> void main() { float a,b,c,D,r1,r2,n,img; printf("Enter value of a="); scanf("%f",&a); printf("Enter value of b="); scanf("%f",&b); printf("Enter value of c="); scanf("%f",&c); D=b*b-4*a*c; printf("D=%.2f\n",D); switch(D>0) { case 1: r1=(-b+sqrt(D))/(2*a); r2=(-b-sqrt(D))/(2*a); printf("r1=%.2f \nr2=%.2f\n",r1,r2); break; case 0: switch(D<0) { case 1: r1=r2=-b/(2*a); img=sqrt(-D)/(2*a); printf("r1=r2=%.2f+i%.2f\n",r1,img); break; case 0: r1=r2=-b/(2*a); printf("r1=r2=%.2f\n",r1); } } } </pre>
-------------	--

**Output
Screenshot**

```

C:\Users\Administrator\Desktop\22TIT093\5.2.exe
Enter value of a=1
Enter value of b=2
Enter value of c=3
D=-8.00
r1=r2=-1.00+i1.41

Process returned 18 (0x12)   execution time : 4.360 s
Press any key to continue.

```

Sr No.	Inputs			Root 1	Root 2	Imaginary
	a	b	c			
1.	1	2	3	-1.00	-1.00	1.41
2.	3	-7	-5	2.91	-0.57	-
3.	9	12	4	-0.67	-0.67	-

Question:

1. Have you learned about how to use normal switch case and nested switch case?

Ans: - In switch statement, variables are tested against a list of values (Case:)for equality. While, in nested switch statement one can use one switch inside another switch.

2. Is default case necessary for every switch case?

Ans: - No it is not necessary of default case in a switch statement and there is no rule of keeping default case at the end of all cases it can be placed at the starting and middle of all other cases.

3. What if break statement is not mentioned between two consecutive cases?

Ans: - If break statement is not mentioned between two consecutive cases then if the first case is true then both the consecutive cases will be executed as break is not mentioned.

Practical 5.3	If the ages of Ram, Shyam and Ajay are input through the keyboard, write a program to determine the youngest of the three. If all of them are of same age then print that “All are of same age”. (Hint: Use Nested if else statement)
Algorithm	<p>Step 1: Start.</p> <p>Step 2: Enter age of Ram, Shyam and Ajay.</p> <p>Step 3: If ram = shyam then check whether shyam = ajay, True – Print all are having same age. False – Print ram and shyam have different age.</p> <p>Step 4: If ram = Ajay, print ram and ajay have same age Else if Shyam = Ajay then prints Shyam and ajay have same age.</p> <p>Step 5: if (ram > shyam) print shyam is younger, Else if (shyam > ajay) Print Ajay is younger Else if (ajay > ram) print ram is younger.</p> <p>Step 6: Stop.</p>
Flow chart	<pre> graph TD Start([Start]) --> Input[/Input r,s,a/] Input --> R1{r==s} R1 --> R1L{ } R1 --> R1R{ } R1L --> S1{s==a} S1 --> Print1[/Print ram==shyam/] Print1 --> AllSame1[/All age are same/] AllSame1 --> End([End]) R1R --> R2{r==a} R2 --> Print2[/Print ram and ajay have same age/] Print2 --> AllDiff1[/All have different age/] AllDiff1 --> End R2 --> S2{s==a} S2 --> Print3[/Print shyam and ajay have same age/] Print3 --> End S2 --> AllDiff2[/All have different age/] AllDiff2 --> End </pre>

Code	<pre> #include<stdio.h> main() { int ram,shyam,ajay; printf("Enter the age of ram:"); scanf("%d",&ram); printf("\nEnter the age of shyam:"); scanf("%d",&shyam); printf("\nEnter the age of ajay:"); scanf("%d",&ajay); if (ram==shyam) { if(shyam==ajay) { printf("\nAll are having same age"); } else { printf("\nram and ajay are having same age"); } } else { if(ram==ajay) { printf("\nram and ajay are having same age"); } else { if(shyam==ajay) { printf("\nshyam and ajay are having same age"); } else { if(ram>shyam) { printf("\nShyam is younger"); } else { if(shyam>ajay) { printf("\najay is younger"); } } } } } } </pre>
-------------	--

	<pre>else { if(ajay>ram) { printf("\nram is younger"); } } }</pre>																																			
Output Screenshot	<div><pre>Enter the age of ram:20 Enter the age of shyam:20 Enter the age of ajay:20 All are having same age Process returned 0 (0x0) execution time : 19.422 s Press any key to continue.</pre></div> <table><tr><th>Sr. N0</th><th colspan="3">Input</th><th>Output</th></tr><tr><th></th><th>Ram</th><th>Shyam</th><th>Ajay</th><th></th></tr><tr><td>1</td><td>12</td><td>12</td><td>12</td><td>SAME</td></tr><tr><td>2</td><td>14</td><td>5</td><td>21</td><td>SHYAM ISYOUNGER</td></tr><tr><td>3</td><td>30</td><td>30</td><td>34</td><td>Ram and Shyam equal</td></tr><tr><td>4</td><td>21</td><td>33</td><td>33</td><td>Shyam and ajay equal</td></tr><tr><td>5</td><td>45</td><td>55</td><td>45</td><td>Ram and ajay equal</td></tr></table>	Sr. N0	Input			Output		Ram	Shyam	Ajay		1	12	12	12	SAME	2	14	5	21	SHYAM ISYOUNGER	3	30	30	34	Ram and Shyam equal	4	21	33	33	Shyam and ajay equal	5	45	55	45	Ram and ajay equal
Sr. N0	Input			Output																																
	Ram	Shyam	Ajay																																	
1	12	12	12	SAME																																
2	14	5	21	SHYAM ISYOUNGER																																
3	30	30	34	Ram and Shyam equal																																
4	21	33	33	Shyam and ajay equal																																
5	45	55	45	Ram and ajay equal																																

Practical 5.4	<p>The policy followed by a company to process customer orders is given by the</p> <p>following rules: Suppose stock=100</p> <p>a) If a customer order is less than or equal to that in stock and 'has credit' is OK, supply 'has requirements.</p> <p>b) If 'has credit' is not OK do not supply. Send him intimation.</p> <p>c) If 'has credit' is OK but the item in stock is less than 'has ordered', inform 'out of stock' and intimate him that the balance will be refunded.</p> <p>Write a C program to implement the company policy.</p>
Algorithm	<p>Step 1: Start.</p> <p>Step 2: Enter value of Credit(a), Order(b), Stock(c).</p> <p>Step 3: If a=Y/y then If a<c then print supply to customer. Else print out of stock.</p> <p>Step 4: else If a=N/n then print not supply Else print enter correct amount.</p> <p>Step 5: Stop</p>
Flow chart	<pre> graph TD Start([Start]) --> Input[/Enter value of credit order and stock/] Input --> D1{if(a==y a==Y)} D1 -- True --> D2{if b<c} D1 -- False --> D3{if(a==n a==N)} D2 -- True --> Output1[/Supply to customer/] D2 -- False --> Output2[/Out of stock/] D3 -- False --> Input2[/Enter correct value/] D3 -- True --> Output3[/Not supply/] Output1 --> End([End]) Output2 --> End Input2 --> End Output3 --> End </pre>

Code	<pre>#include<stdio.h> main() { char a; int b,c; printf("Enter value of credit:"); scanf("%c",&a); printf("\nEnter value of order:"); scanf("%d",&b); printf("\nEnter value of stock:"); scanf("%d",&c); if(a=='Y' a=='y') { if(b<c) { printf("\nSupply the stock to customer"); } else { printf("\nOut of stock"); } } else { if(a=='n' a=='N') { printf("\nnot supply"); } else { printf("\nPlease enter correct amount"); } } }</pre>
-------------	---

**Output
Screenshot**

```

Enter value of credit:y

Enter value of order:12

Enter value of stock:20

Supply the stock to customer
Process returned 0 (0x0)   execution time : 9.815 s
Press any key to continue.

```

Sr. No.	Inputs			Output
	Credit	Order	Stock	
1	Y or y	20	100	Supply
2	N or n	50	80	Not supply
3	Y or y	50	80	Supply
4	Y or y	70	30	Out of stock
5	Y or y	30	30	Out of stock

Question:

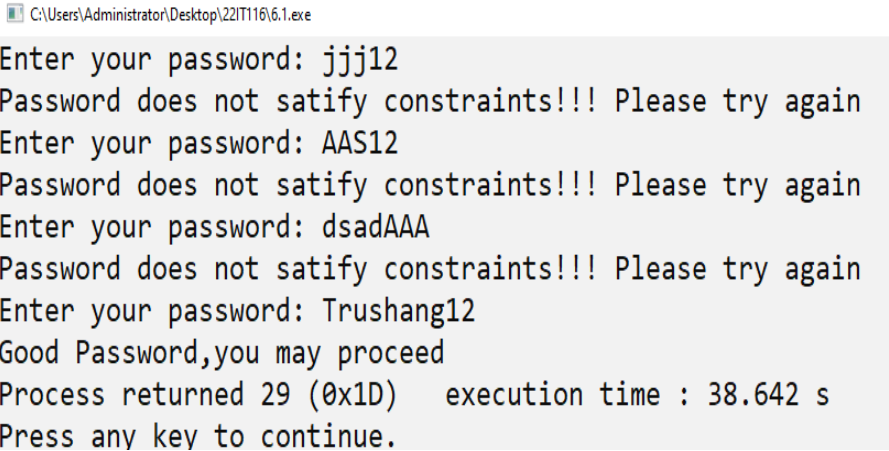
1. Which kind of logic have you used for building this program?
If else if ladder or nested if else statements?

Ans: I have used nested if else statements for building this program.

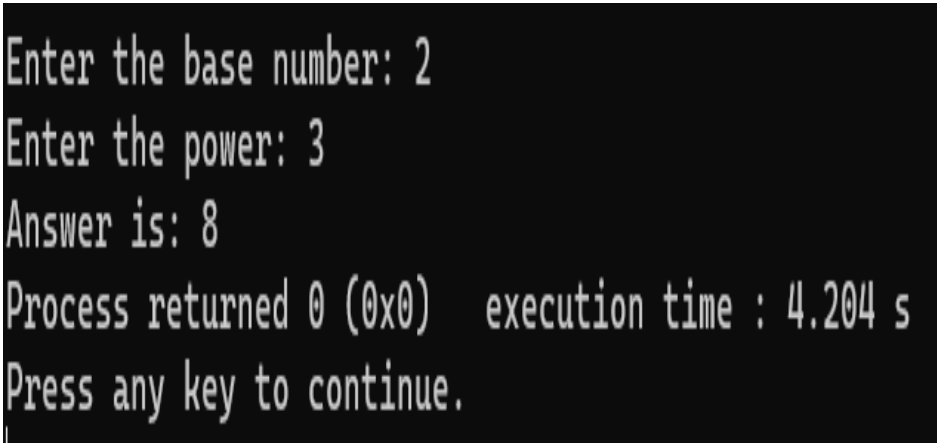
Sign:**Grade:**

	SET – 6 : Looping
Practical 6.1	<p>There is a person, who is asked to enter the alphanumeric password for registering into an ecommerce website for purchasing products from website. But he is not aware about, what does Alphanumeric mean. So, he tries entering various combinations 5 times, but he fails to create such password. So let us help him by writing a C program to validate his password. Constraints for writing password are it should have combination of lowercase, uppercase and digit.</p> <p>Note: Use Do while loop, and give print appropriate outputs on incorrect validations.</p> <p>Expected Outcome: Draw flowchart and write algorithm and write program for given scenario. Mention all the inputs that you have experimented and outputs received. Also mention the correct alphanumeric password created by you.</p>
Algorithm	<p>Step 1: Start.</p> <p>Step 2: Declare character array s.</p> <p>Step 3: Input s.</p> <p>Step 4: If password contains both alphabets and number print password is alphanumeric.</p> <p>Step 5: Else print password is not alphanumeric and go to step 3.</p> <p>Step 6: End.</p>
Flow chart	<pre> graph TD Start([Start]) --> Input[/Input s/] Input --> Decision{Password contains lowercase, uppercase and digit} Decision -- True --> Correct[/Correct Password/] Correct --> End([End]) Decision -- False --> Incorrect[/Incorrect password/] Incorrect --> Input </pre>

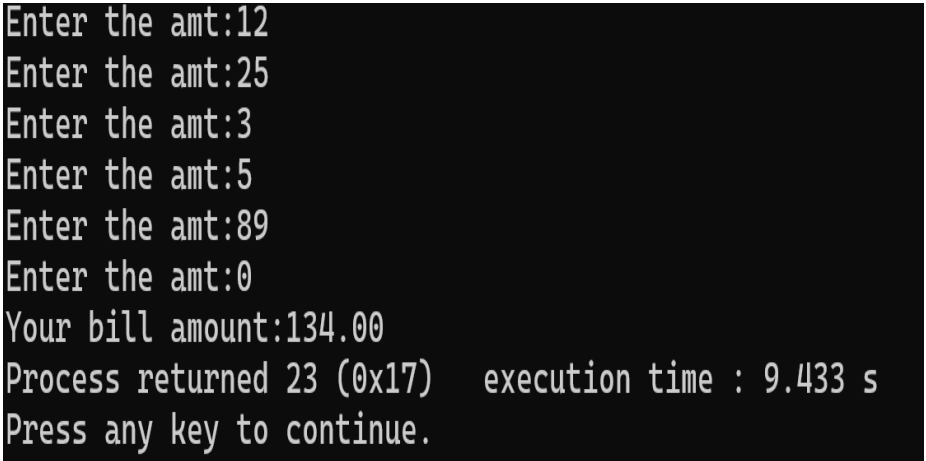
Code	<pre>#include<ctype.h> void main() { char s[10]; int i,l,u,d,a; L: i=0,l=0,u=0,d=0,a=0; printf("Enter your password: "); scanf("%s",&s); do { if(isupper(s[i])>0) { u++; } else if(islower(s[i])>0) { l++; } else if(isdigit(s[i])>0) { d++; } else { a++; } i++; }while(s[i]); if(u+l+d==i&&u!=0&&l!=0&&d!=0) { printf("Good Password,you may proceed"); } else { printf("Password does not satisfy constraints!!! Please try again\n"); goto L; } }</pre>
-------------	--

Output Screenshot	<div></div> <table><thead><tr><th>Sr.No.</th><th>Input</th><th>Output</th></tr></thead><tbody><tr><td>1</td><td>AAAS12</td><td>Password does not satisfy constraints!!! Please try again.</td></tr><tr><td>2</td><td>Trushang12</td><td>Good password, you may proceed</td></tr></tbody></table>	Sr.No.	Input	Output	1	AAAS12	Password does not satisfy constraints!!! Please try again.	2	Trushang12	Good password, you may proceed
Sr.No.	Input	Output								
1	AAAS12	Password does not satisfy constraints!!! Please try again.								
2	Trushang12	Good password, you may proceed								
Question:	<div><div><div>1. Have you understood working of do...while loop? Do mention the syntax of this loop.</div><div>Ans: Yes, I understood the working of do...while loop. Syntax of do...while loop: <pre>do { statements } while(condition);</pre></div></div><div><div>2. Have you used loop in this program?</div><div>Ans: Yes, I have used do...while loop in this program.</div></div><div><div>3. What is goto statement? How it is useful?</div><div>Ans: The goto statement is a jump statement which is sometimes also referred to as unconditional jump statement. The goto statement can be used to jump from anywhere to anywhere within a function. The goto statement can be used to alter the flow of control in a program.</div></div></div>									

Practical 6.2	Two numbers are entered through the keyboard. Write a program to find the value of one number raised to the power of another. (Use While loop)
Algorithm	<p>Step 1: Start</p> <p>Step 2: Input base number b and power number p.</p> <p>Step 3: Declare counter i=1 and ans = 1.</p> <p>Step 4: Run loop up till $i \leq p$</p> <p style="padding-left: 40px;">ans = ans*b</p> <p style="padding-left: 40px;">i++</p> <p>Step 5: Print ans.</p> <p>Step 6: End.</p>
Flow chart	<pre> graph TD Start([Start]) --> Input[/Input b, p i=1, ans=1/] Input --> Process[ans=ans*b i++] Process --> Decision{i <= p} Decision -- True --> Process Decision -- False --> Output[/Print ans/] Output --> End([End]) </pre>

Code	<pre>#include<Stdio.h> main() { int ans=1,i=1,p,b; printf("Enter the base number: "); scanf("%d",&b); printf("Enter the power: "); scanf("%d",&p); while(i<=p) { ans = ans*b; i++; } printf("Answer is: %d",ans); }</pre>																
Output Screenshot	<div></div> <table><tr><th>Sr. No</th><th>Base No.</th><th>Power</th><th>Output</th></tr><tr><td>1</td><td>3</td><td>2</td><td>9</td></tr><tr><td>2</td><td>5</td><td>3</td><td>125</td></tr><tr><td>3</td><td>2</td><td>3</td><td>8</td></tr></table>	Sr. No	Base No.	Power	Output	1	3	2	9	2	5	3	125	3	2	3	8
Sr. No	Base No.	Power	Output														
1	3	2	9														
2	5	3	125														
3	2	3	8														
Question:	<p>1. Have you understood the concept of while loop? if yes write its syntax here.</p> <p>Ans: Yes, I understood the concept of while loop.</p> <p>Syntax of while loop:</p> <pre>while(condition) { statements; }</pre>																

Practical 6.3	Write a C program for big bazaar cashier to count the amount to be collected from the customer. Cashier will enter the numbers one after another for each item and to get the summation of entered numbers, he has to enter 0. (Use for loop) (Hint: Break statement can be used)
Algorithm	<p>Step 1: Start.</p> <p>Step 2: Input i, q, p, sum=0.</p> <p>Step 3: Enter item number i.</p> <p>Step 4: If i>0 Enter quantity of of item q. Enter price of item p. sum = sum+(p*q) goto step 3.</p> <p>Step 5: Else if i==0, break.</p> <p>Step 6: End.</p>
Flowchart	<pre> graph TD Start([Start]) --> Input[/Input amt sum=0/] Input --> Decision{amt!=0} Decision -- True --> Process[Input amt sum=sum + amt] Process --> Decision Decision -- False --> Output[/Print total bill=sum/] Output --> End([End]) </pre>

Code	<pre>#include<stdio.h> void main() { float amt,sum=0; for(;amt!=0;) { printf("Enter the amt:"); scanf("%f",&amt); sum=sum+amt; } printf("Your bill amount:%.2f",sum); }</pre>
Output Screenshot	 <pre>Enter the amt:12 Enter the amt:25 Enter the amt:3 Enter the amt:5 Enter the amt:89 Enter the amt:0 Your bill amount:134.00 Process returned 23 (0x17) execution time : 9.433 s Press any key to continue.</pre>
Question:	<p>1. Have you learned the concept of for loop using above given scenario? Explain what does ‘i’ stands for in the for () loop, consider the given example below. e.g. for (i=0; i<10; i++)</p> <p>Ans: Yes, I have learned the concept of for loop from the above given scenario. Here i stands for the counter till which the loop is going to be executed.</p>

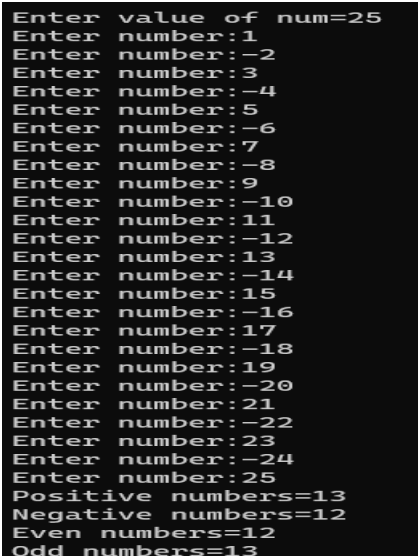
Practical 6.4	<p>Write a program for a match-stick game between the computer and a user.</p> <p>Your Program should ensure that the computer always wins. Rules for the games are as follows:</p> <ul style="list-style-type: none"> • There are 21 match-sticks. • The computer asks the player to pick 1, 2, 3, or 4 match-sticks. • After the person picks, the computer does its picking. • Whoever is forced to pick up the last match-stick loses the game. <p>Use while loop, break and Continue Statements.</p>
Algorithm	<p>Step 1: Start</p> <p>Step 2: Declare n=21, y, c, i.</p> <p>Step 3: use of while loop</p> <p>Input y.</p> <p>Step 4: If $y < 1 \parallel y > 4$</p> <p>Please enter valid choice</p> <p>Else</p> <p>$n = n - y$ and print n.</p> <p>$c = 5 - y$ and print c.</p> <p>$n = n - c$ and print n.</p> <p>Step 5: if $n == 1$</p> <p>Print You loose the game.</p> <p>Step 6: End.</p>
Flowchart	<pre> graph TD Start([Start]) --> Declare[/Declare n=21 y,c,i/] Declare --> While{while(1)} While -- True --> Input[/Input y/] Input --> Validity{y < 1 y > 4} Validity -- True --> PrintValid[/Print valid choice/] PrintValid --> While Validity -- False --> Process[n=n-y print n c=5-y print c n=n-c print n] Process --> N1{n==1} N1 --> PrintLose[/print You loose the game break/] PrintLose --> End([End]) </pre>

Code	<pre> void main() { int n=21,y,c,i; while(1) { printf("User pick matchstick:"); scanf("%d",&y); if(y<1 y>4) { printf("Plese enter vaild choice\n"); continue; } else { n=n-y; printf("Matchstick=%d\n",n); c=5-y; printf("Computer Pick matchstick :%d\n",c); n=n-c; printf("Matchstick=%d\n",n); if(n==1) { printf("You lose the game"); break; } } } } </pre>
Output Screenshot:	<pre> C:\Users\Administrator\Desktop\6.4.exe User pick matchstick:4 Matchstick=17 Computer Pick matchstick :1 Matchstick=16 User pick matchstick:3 Matchstick=13 Computer Pick matchstick :2 Matchstick=11 User pick matchstick:2 Matchstick=9 Computer Pick matchstick :3 Matchstick=6 User pick matchstick:1 Matchstick=5 Computer Pick matchstick :4 Matchstick=1 You lose the game Process returned 17 (0x11) execution time : 4.159 s Press any key to continue. </pre>

	Sr.No.	Entered Number by User	Entered Number by Computer	Sticks left
	1	4	1	16
	2	3	2	11
	3	2	3	6
	4	1	4	1
Question:	1. What is the significance of using break and continue statement?			
	Ans: The break statement is used to terminate the loop immediately. The continue statement is used to skip the current iteration of the loop.			

Sign:**Grade:**

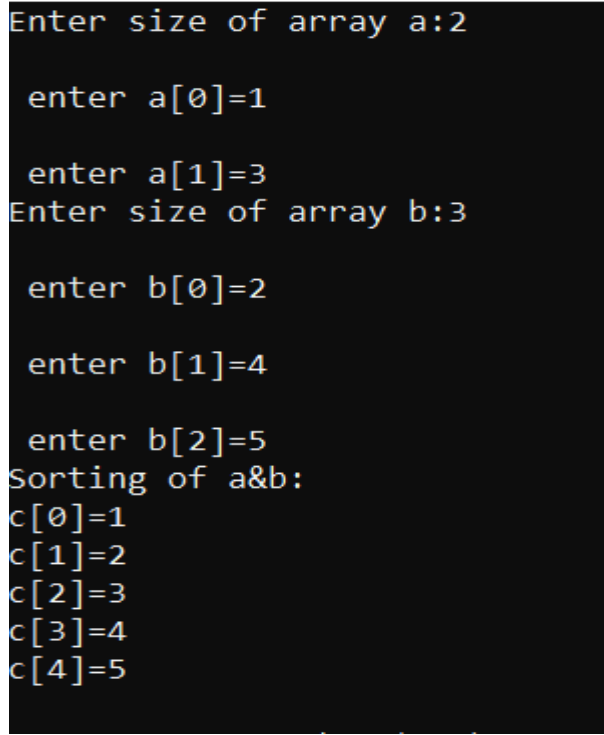
	SET-7: Arrays
Practical 7.1	Twenty-five numbers are entered from the keyboard into an array. Write a C program to find out how many numbers of them are positive, negative, and how many are even and odd?
Algorithm	<p>Step 1:Start</p> <p>Step 2:Declare a[25],n=p=e=o=0.</p> <p>Step 3:Using for loop scan the array from user.</p> <p>Step 4:if a[i]%2= 0 its even number. Else its odd number.</p> <p>Step 5:if a[i]<0 its negative, Else its positive.</p> <p>Step 6:Print even,odd,positive,negative numbers.</p>
Flowchart	<pre> graph TD Start([Start]) --> Input[/Input array a[25] i=e=o=p=n=0/] Input --> Cond1{a[i]%2==0} Cond1 -- True --> Einc[e++] Cond1 -- False --> Oinc[o++] Einc --> Cond2{a[i]>0} Oinc --> Cond2 Cond2 -- True --> Pinc[p++] Cond2 -- False --> Ninc[n++] Pinc --> Iinc[i++] Ninc --> Iinc Iinc --> Cond3{i<25} Cond3 -- True --> Cond1 Cond3 -- False --> Print[/Print n,p,e,o/] Print --> End([End]) </pre> <p>The flowchart starts with an oval 'Start' leading to a parallelogram 'Input array a[25] i=e=o=p=n=0'. It then enters a loop. The first decision diamond is 'a[i]%2==0'. If 'True', it goes to a rectangle 'e++'. If 'False', it goes to a rectangle 'o++'. Both paths lead to a second decision diamond 'a[i]>0'. If 'True', it goes to a rectangle 'p++'. If 'False', it goes to a rectangle 'n++'. Both paths lead to a rectangle 'i++'. This leads to a third decision diamond 'i<25'. If 'True', it loops back to the 'a[i]%2==0' diamond. If 'False', it goes to a parallelogram 'Print n,p,e,o', which then leads to an oval 'End'.</p>

Code	<pre> void main() { int i,a[25],num,p=0,n=0,e=0,o=0; printf("Enter value of num="); scanf("%d",&num); for(i=0;i<num;i++) { printf("Enter number:"); scanf("%d",&a[i]); if(a[i]%2==0) { e++; } else { o++; } if(a[i]>0) { p++; } else { n++; } } printf("Positive numbers=%d\n",p); printf("Negative numbers=%d\n",n); printf("Even numbers=%d\n",e); printf("Odd numbers=%d\n",o); } </pre>
Output screenshot	 <pre> Enter value of num=25 Enter number:1 Enter number:-2 Enter number:3 Enter number:-4 Enter number:5 Enter number:-6 Enter number:7 Enter number:-8 Enter number:9 Enter number:-10 Enter number:11 Enter number:-12 Enter number:13 Enter number:-14 Enter number:15 Enter number:-16 Enter number:17 Enter number:-18 Enter number:19 Enter number:-20 Enter number:21 Enter number:-22 Enter number:23 Enter number:-24 Enter number:25 Positive numbers=13 Negative numbers=12 Even numbers=12 Odd numbers=13 </pre>

	Sr.No.	Parameter	Counts
	1	Positive number	13
	2	Negative number	12
	3	Even number	12
	4	Odd number	13
Question	<p>1. Is it necessary to initialize a variable with zero every time? If yes, then why? If no, then when is it necessary to initialize the number with zero and why?</p> <p>Ans: Yes, it is necessary to initialize a variable with zero every time because if we don't initialize system would take a garbage value for that variable.</p>		

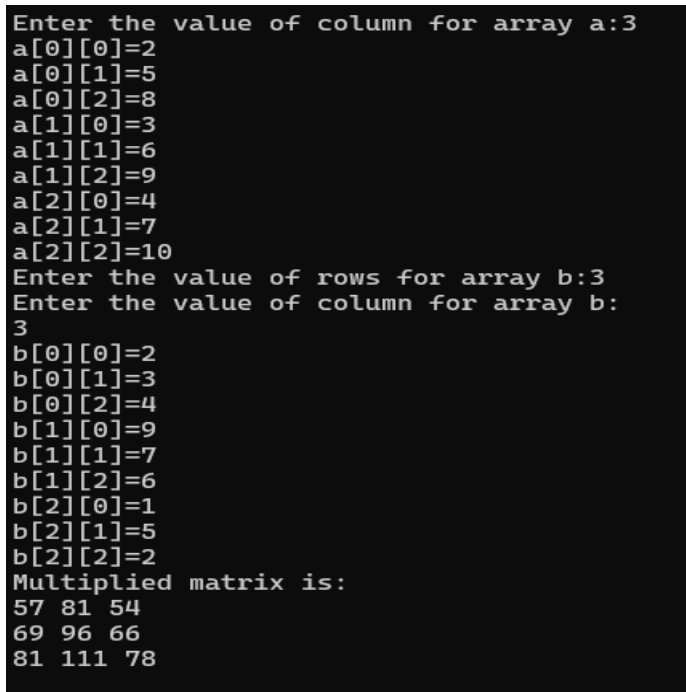
Practical 7.2	Write a program for creating two arrays of different size and merge both arrays into one by sorting those arrays in ascending order. [Merge by sorting]
Algorithm	<p>Step 1: Start</p> <p>Step 2: Input n ,array a, b , and declare m , i , j ,k</p> <p>Step 3:if $a[i] > b[j]$ then $c[k] = b[j]$ $j++$, $k++$ Else $c[k] = a[i]$ $i++$, $k++$</p> <p>Step 4: In while loop $i < n$ then $c[k] = a[i]$ $i++$, $k++$</p> <p>Step 5: In while loop $j < m$ then $c[k] = b[j]$ $j++$, $k++$</p> <p>Step 6: Print $c[k]$</p> <p>Step 7: End.</p>
Flowchart	<pre> graph TD Start([Start]) --> Input[/Input n, array a, b Declare m, i, j, k/] Input --> Cond1{a[i] > b[j]} Cond1 -- True --> Process1[c[k] = b[j] j++ k++] Cond1 -- False --> Process2[c[k] = a[i] i++ k++] Process1 --> Cond2{while loop j < n} Process2 --> Cond2 Cond2 -- True --> Process3[c[k] = a[i] i++ k++] Cond2 -- False --> Cond3{While loop j < m} Process3 --> Cond3 Cond3 -- True --> Process4[c[k] = b[j] j++ k++] Cond3 -- False --> Print[Print c[k]] Process4 --> Print Print --> End([End]) </pre>

Code	<pre> void main() { int a[10],b[10],c[10],n,m,i,j,k; printf("Enter size of array a:"); scanf("%d",&n); for(i=0;i<n;i++) { printf("\n enter a[%d]=",i); scanf("%d",&a[i]); } printf("Enter size of array b:"); scanf("%d",&m); for(j=0;j<m;j++) { printf("\n enter b[%d]=",j); scanf("%d",&b[j]); } for(i=0,j=0,k=0;i<n&& j<m;) { if(a[i]>b[j]) { c[k]=b[j]; j++; k++; } else { c[k]=a[i]; i++; k++; } } while(i<n) { c[k]=a[i]; i++; k++; } while(j<m) { c[k]=b[j]; j++; k++; } printf("Sorting of a&b:\n"); </pre>
-------------	--

	<pre>for(k=0;k<(n+m);k++) { printf("c[%d]=%d\n",k,c[k]); } }</pre>
Output screenshot	 <p>The screenshot shows the following text on a black background with yellow and white text:</p> <pre>Enter size of array a:2 enter a[0]=1 enter a[1]=3 Enter size of array b:3 enter b[0]=2 enter b[1]=4 enter b[2]=5 Sorting of a&b: c[0]=1 c[1]=2 c[2]=3 c[3]=4 c[4]=5</pre>

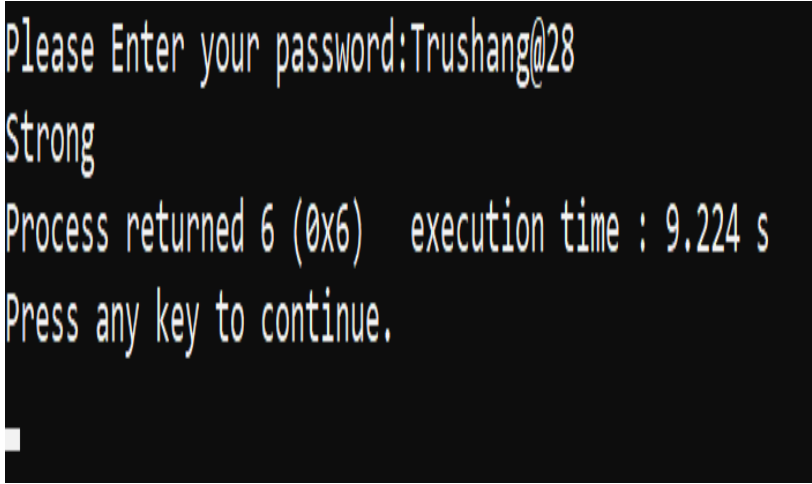
Practical 7.3	<p>Write a program to multiply any two 3*3 matrices.</p> <p>Test Data:</p> <p>Input the rows and columns of first matrix : 3 3</p> <p>Input the rows and columns of second matrix : 3 3</p>
Algorithm	<p>Step 1: Start</p> <p>Step 2: Declare 2-d array (a,b,c),r1,c1,r2,c2,i,j,k</p> <p>Step 3: Input r1 ,c1 ,a[i][j],c[i][j]=0;r2,c2</p> <p>Step 4: If c1==r2 then input b[i][j] and c[i][j]=c[i][j]+(a[i][k]*b[k][i])</p> <p>Else Print Multiply not possible and goto step3</p> <p>Step 4: Print c[i][j]</p> <p>Step 5: End</p>
Flowchart	<pre> graph TD Start([Start]) --> Declare[/Declare 2-d array a, b, c r1, c1, r2, c2, i, j, k/] Declare --> Input[/Input r1, c1, r2, c2 a[i][j], c[i][j]=0/] Input --> Decision{c1==r2} Decision -- True --> Process[input b[i][j] c[i][j]=c[i][j]+(a[i][k]*b[k][i])] Decision -- False --> PrintNotPossible[Print Multiply not possible] Process --> PrintC[/Print c[i][j]/] PrintNotPossible --> PrintC PrintC --> End([End]) </pre>

Code	<pre> void main() { int a[10][10],b[10][10],c[10][10],r1,c1,r2,c2,i,j,k; L: printf("Enter the value of rows for array a:"); scanf("%d",&r1); printf("Enter the value of column for array a:"); scanf("%d",&c1); for(i=0;i<r1;i++) { for(j=0;j<c1;j++) { c[i][j]=0; } } for(i=0;i<r1;i++) { for(j=0;j<c1;j++) { printf("a[%d][%d]=",i,j); scanf("%d",&a[i][j]); } } printf("Enter the value of rows for array b:"); scanf("%d",&r2); printf("Enter the value of column for array b:"); scanf("%d",&c2); if(c1==r2) { for(i=0;i<r1;i++) { for(j=0;j<c1;j++) { printf("b[%d][%d]=",i,j); scanf("%d",&b[i][j]); } } for(i=0;i<r1;i++) { for(j=0;j<c2;j++) { for(k=0;k<c1;k++) { c[i][j] = c[i][j] + (a[i][k]*b[k][j]); } } } } } </pre>
-------------	--

	<pre> printf("Multiplied matrix is: "); printf("\n"); } else { printf("Multiply of two Matrices is not possible\n"); goto L; } for(i=0;i<r1;i++) { for(j=0;j<c2;j++) { printf("%d ",c[i][j]); } printf("\n"); } } </pre>
Output screenshot	 <pre> Enter the value of column for array a:3 a[0][0]=2 a[0][1]=5 a[0][2]=8 a[1][0]=3 a[1][1]=6 a[1][2]=9 a[2][0]=4 a[2][1]=7 a[2][2]=10 Enter the value of rows for array b:3 Enter the value of column for array b: 3 b[0][0]=2 b[0][1]=3 b[0][2]=4 b[1][0]=9 b[1][1]=7 b[1][2]=6 b[2][0]=1 b[2][1]=5 b[2][2]=2 Multiplied matrix is: 57 81 54 69 96 66 81 111 78 </pre>
Question-Answers	<p>1. State the advantages of using Array Indexes. When is it suitable to take array index?</p> <p>Ans: In arrays, the elements can be accessed randomly by using the index number. Arrays allocate memory in contiguous memory locations for all its elements. Hence there is no chance of extra memory being allocated in case of arrays. This avoids memory overflow or shortage of memory in arrays.</p>

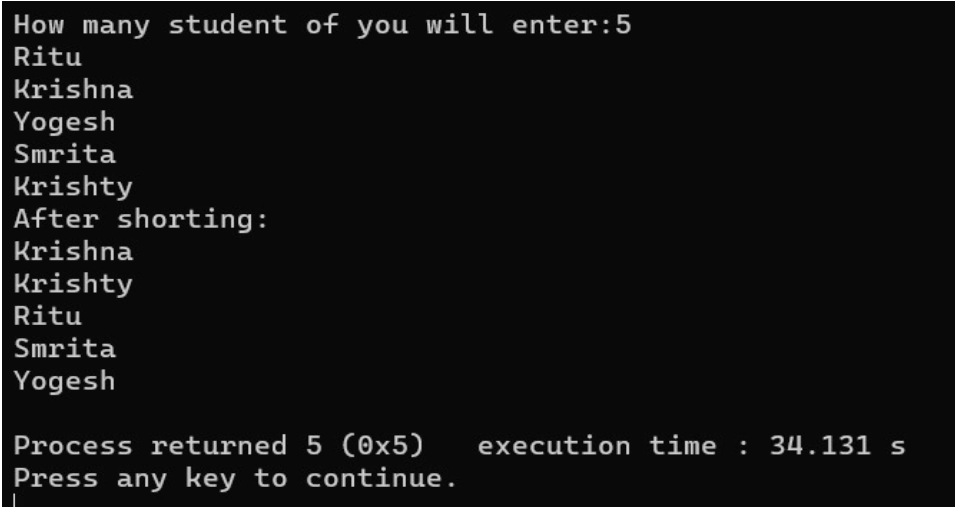
Sign:**Grade:**

	SET-8: Character Arrays and Strings
Practical 8.1	<p>Help user to identify how strong is his password based on the number of lowercase, alphabets, uppercase alphabets, digits and special characters given by the user from the keyboard. Length of entered password(string) should be of 8.</p> <p>Constraints for identifying strength of password:</p> <ol style="list-style-type: none"> Strong: Mixture of lowercase alphabets, uppercase alphabets, digits and special characters Average: Mixture of lowercase alphabets, digits and special characters Poor: Either only has alphabets/digits/special characters
Algorithm	<p>Step 1:Start</p> <p>Step 2:Input password and declare i=u=n=i=0</p> <p>Step 3:if password contain uppercase then u=1 Else if password contain lowercase, then l=1 Else if password contain digit, then n=1 Else password contains special symbol then s=1</p> <p>Step 4:if l==1&& u==1&&n==1&&s==1 then print strong Else if l==1 &&n==1&&s==1 then print average Else then print poor</p> <p>Step 5: End</p>
Flowchart	<pre> graph TD Start([Start]) --> Input[/Input password Declare i=u=n=i=0/] Input --> Loop{i < strlen(pass)} Loop --> D1{password contain spacial symbol} Loop --> D2{password contain digit} Loop --> D3{password contain lowercase} Loop --> D4{password contain uppercase} D1 -- True --> S1[s=1] D1 -- False --> J1(()) D2 -- True --> S2[n=1] D2 -- False --> J1 D3 -- True --> S3[l=1] D3 -- False --> J1 D4 -- True --> S4[u=1] D4 -- False --> J1 J1 --> D5{l==1 & u==1 n==1 & s==1} S1 --> D5 S2 --> D5 S3 --> D5 S4 --> D5 D5 -- True --> PStrong[/Print Strong/] D5 -- False --> D6{l==1 & n==1 & s==1} D6 -- True --> PAverage[/Print Average/] D6 -- False --> PPoor[/Print poor/] PStrong --> End([End]) PAverage --> End PPoor --> End </pre>

Code	<pre> void main() { char pass[100]; printf("Please Enter your password:"); gets(pass); int l=0,u=0,n=0,s=0; for(int i=0;i<strlen(pass);i++) { if(pass[i]>='a' && pass[i]<='z') l=1; else if(pass[i]>='A' && pass[i]<='Z') u=1; else if(pass[i]>='0' && pass[i]<='9') n=1; else s=1; } if(l==1 && u==1 && n==1 && s==1) printf("Strong"); else if(l==1 && n==1 && s==1) printf("Average"); else printf("Poor"); } </pre>
Output screenshot	

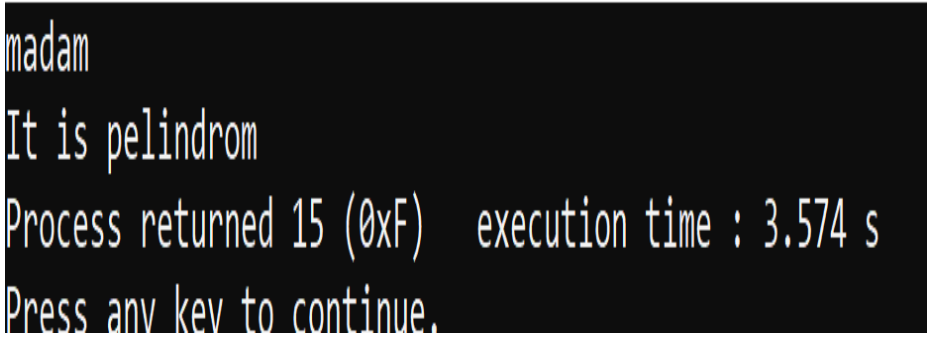
	Sr. No.	Input	Lowercase	Uppercase	Numbers	Symbols
	1	Trushan@2	Yes	Yes	Yes	Yes
	2	trushang@1	Yes	No	Yes	Yes
	3	123456	No	No	Yes	No
Question	<p>1. Explain the difference between string and character. Also write the syntax for printing character and string.</p> <p>Ans: The main difference between Character and String is that Character refers to a single letter, number, space, punctuation mark or a symbol that can be represented using a computer while String refers to a set of characters. In C programming, we can use char data type to store both character and string values.</p>					

Practical 8.2	Let us assume, teacher is supposed to allot seats based on the student's names. You are requested to help teacher by creating a C program, for collecting the names of 5 students and sort them in alphabetical order.
Algorithm	Step 1: Start Step 2: Input 5 string in a array Step 3:compare two strings respectively Step 4:Short the array in alphabetical order Step 5:Print the sorted array Step 6:End
Flowchart	<pre>graph TD; Start([Start]) --> Input[/Input 5 string in a array/]; Input --> Compare[compare two string respectively]; Compare --> Sort[Short the array in alphabetical order]; Sort --> Print[/Print the sorted array/]; Print --> End([End]);</pre>

Code	<pre> #include<string.h> void main() { char s[10][10],t[10]; int i,j,n; printf("How many student of you will enter:"); scanf("%d",&n); for(i=0;i<n;i++) scanf("%s",s[i]); for(i=0;i<n-1;i++) { for(j=i+1;j<n;j++) { if(strcmp(s[i],s[j])>0) { strcpy(t[i],s[i]); strcpy(s[i],s[j]); strcpy(s[j],t[i]); } } } printf("After shorting:"); for(i=0;i<n;i++) { printf("%s\n",s[i]); } } </pre>
Output screenshot	 <p>How many student of you will enter:5 Ritu Krishna Yogesh Smrita Krishty After shorting: Krishna Krishty Ritu Smrita Yogesh</p> <p>Process returned 5 (0x5) execution time : 34.131 s Press any key to continue.</p>

	Sr. No.	Input of names	Sorted output
	1	Ritu	Krishna
	2	Krishna	Krishty
	3	Yogesh	Ritu
	4	Smrita	Smrita
	5	Krishty	Yogesh
Question	1. Which string functions have you learned from this program? Explain any 5 string functions in below given table.		
	Ans:		
	Sr.No.	String function syntax	Purpose
	1	strlen(string-name)	Returns the length of string -name.
	2	Strcpy(destination , source)	Copies the contents of source string to destination string
	3	strcat(first_string,second_string)	concat or joins first string with second string. The result of the string is stored in first string.
	4	strrev(string)	returns the reversed string.
	5	strlwr(string)	return the string characters in lowercase

Practical 8.3	Write a C program to check if the user inputted string is palindrome or not using recursion
Algorithm	<p>Step 1:Start</p> <p>Step 2:Input the string ,flag=1,l,i=0</p> <p>Step 3:if i<2</p> <p>Step 4:if s[i]!=s[l-i-1] then flag=1</p> <p>Step 5: if flag==0 then Print palindrome Else if Print not palindrome</p> <p>Step 6:End</p>
Flowchart	<pre> graph TD Start([Start]) --> Input[/Input the string flag=1,l,i=0/] Input --> Cond1{i < l/2} Cond1 -- True --> Cond2{s[i] != s[l-i-1]} Cond2 --> SetFlag(flag = 1) SetFlag --> Cond3{flag == 0} Cond3 -- False --> PrintNot[/Print Not pelindrom/] Cond3 -- True --> PrintPal[/Print pelindrom/] PrintNot --> End([End]) PrintPal --> End </pre> <p>The flowchart illustrates the logic for checking a string is a palindrome or not. It begins with a 'Start' terminal, followed by an input process 'Input the string flag=1,l,i=0'. A decision diamond 'i < l/2' checks the loop condition. If 'True', it proceeds to another decision diamond 's[i] != s[l-i-1]'. If this condition is true, it sets 'flag=1'. Then, a decision diamond 'flag == 0' checks if the string is a palindrome. If 'False', it prints 'Print Not pelindrom'. If 'True', it prints 'Print pelindrom'. Both paths lead to the 'End' terminal.</p>

Code	<pre>#include<string.h> void main() { char s[10]; int flag,l; gets(s); l=strlen(s); for(int i=0;i<l/2;i++) { if(s[i]!=s[l-i-1]) { flag=1; break; } } if(flag==0) printf("It is pelindrom"); else printf("It is not pelindrom"); }</pre>															
Output screenshot	<div></div> <table><tr><th>Sr.No.</th><th>Input</th><th>Output</th></tr><tr><td>1</td><td>Alpha</td><td>Not palindrome</td></tr><tr><td>2</td><td>Madam</td><td>Palindrome</td></tr><tr><td>3</td><td>saippuakivikauppias</td><td>Palindrome</td></tr><tr><td>4</td><td>Hannah</td><td>Palindrome</td></tr></table>	Sr.No.	Input	Output	1	Alpha	Not palindrome	2	Madam	Palindrome	3	saippuakivikauppias	Palindrome	4	Hannah	Palindrome
Sr.No.	Input	Output														
1	Alpha	Not palindrome														
2	Madam	Palindrome														
3	saippuakivikauppias	Palindrome														
4	Hannah	Palindrome														

Question

1. Explain the concept of recursion. Explain the difference between recursion and iteration?

Ans: Recursion is the process of defining a problem (or the solution to a problem) in terms of (a simpler version of) itself. For example, we can define the operation "find your way home" as: If you are at home, stop moving. Take one step toward home.

Property	Recursion	Iteration
Definition	Function calls itself.	A set of instructions repeatedly executed.
Application	For functions.	For loops.
Termination	Through base case, where there will be no function call.	When the termination condition for the iterator ceases to be satisfied.
Usage	Used when code size needs to be small, and time complexity is not an issue.	Used when time complexity needs to be balanced against an expanded code size.
Code size	Smaller code size	Larger code size.
Time Complexity	Very high(generally exponential) time complexity.	Relatively lower time complexity(generally polynomial-logarithmic).

Sign:**Grade:**

	SET-9: User-Defined Function in 'C'
Practical 9.1	<p>Write a C program to check if the entered number is prime or not by using types of user defined functions</p> <p>(i) No arguments passed and no return value (ii) No arguments passed but a return value (iii) Argument passed but no return value (iv) Argument passed and a return value</p>
Algorithm	<p>Step 1: Start</p> <p>Step 2: Read number n</p> <p>Step 3: Set f=0</p> <p>Step 4: For i=2 to n-1</p> <p>Step 5: If n mod i=0 then</p> <p>Step 6: Set f=1 and break</p> <p>Step 7: Loop</p> <p>Step 8: If f=0 then print 'The given number is prime' else print 'The given number is not prime'</p> <p>Step 9: End.</p>
Flow chart	<pre> graph TD Start([Start]) --> Input[/Input number n , x=1/] Input --> Cond1{x < n} Cond1 -- True --> Inc[x++] Inc --> Cond1 Cond1 -- False --> Cond2{x/n == 1} Cond2 -- True --> Prime[/Prime number/] Prime --> End([End]) Cond2 -- False --> NotPrime[/Not Prime/] NotPrime --> End </pre>

Code	<pre> void prime2(); void prime1(int a); int prime3(); int prime4(int a); void main() { int n,num,i=2,flag=0; printf("Please Enter number:"); scanf("%d",&n); printf("with no return and with argument:\n"); prime1(n); printf("with no return and with no argument:\n"); prime2(); printf("with return and with no argument:\n"); flag=prime3(); if(flag==1) { printf("Given number is Not prime\n"); } else { printf("Given number is prime\n"); } printf("with return and with argument:\n"); flag=prime4(n); if(flag==1) { printf("Given number is Not prime\n"); } else { printf("Given number is prime\n"); } } void prime1(int n) { int i,flag=0; for(i=2;i<n/2;i++) { if(n%i==0) { flag=1; break; } } if(flag==1) { </pre>
-------------	--

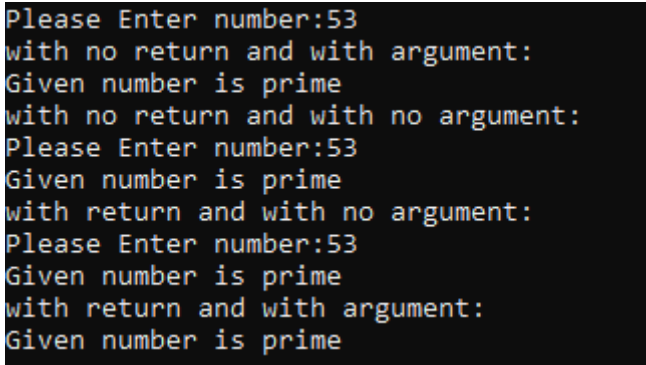
```

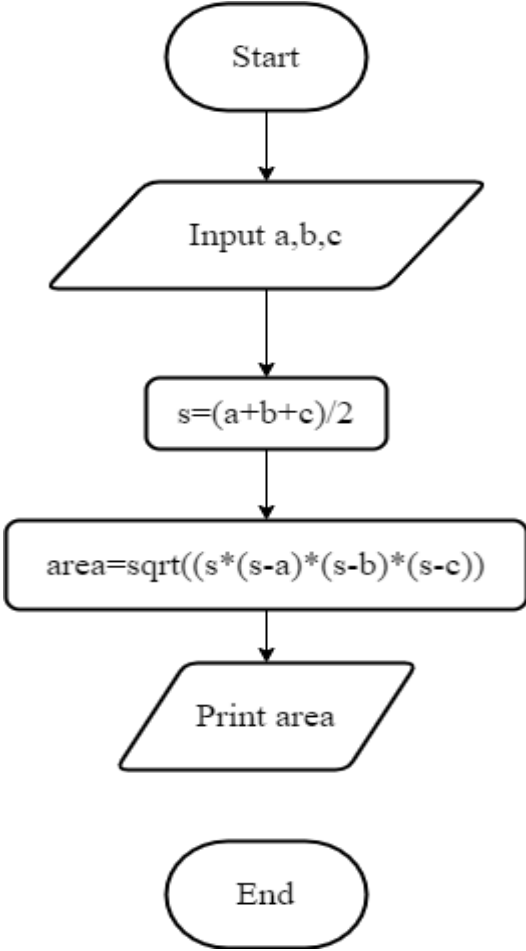
        printf("Given number is Not prime\n");
    }
    else
    {
        printf("Given number is prime\n");
    }
}
void prime2()
{
    int n,i;
    printf("Please Enter number:");
    scanf("%d",&n);

    int flag=0;
    for(i=2;i<n/2;i++)
    {
        if(n%i==0)
        {
            flag=1;
            break;
        }
    }
    if(flag==1)
    {
        printf("Given number is Not prime\n");
    }
    else{
        printf("Given number is prime\n");
    }
}
int prime3()
{
    int n;
    printf("Please Enter number:");
    scanf("%d",&n);

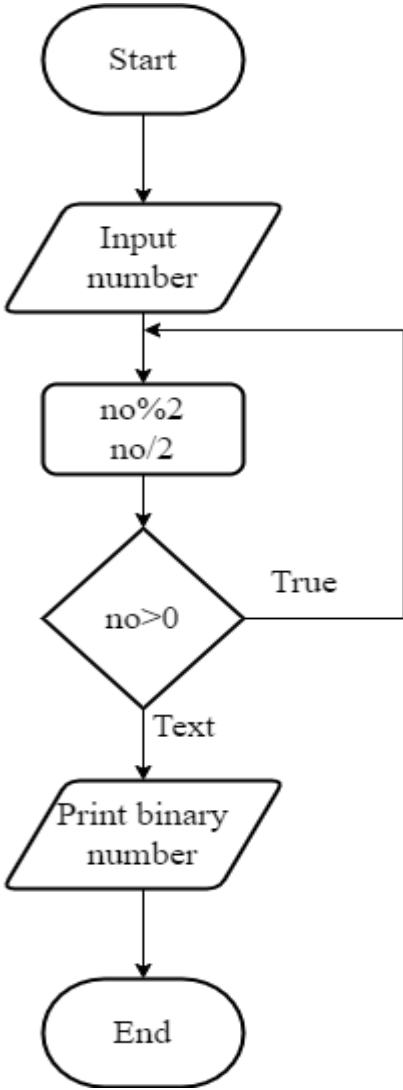
    int flag=0,i;
    for(i=2;i<n/2;i++)
    {
        if(n%i==0)
        {
            flag=1;
            break;
        }
    }
    return flag;
}
int prime4(int n)

```


	<pre> { int i,flag=0; for(i=2;i<n/2;i++) { if(n%i==0) { flag=1; break; } } return flag; } </pre>
Output screenshot	 <pre> Please Enter number:53 with no return and with argument: Given number is prime with no return and with no argument: Please Enter number:53 Given number is prime with return and with no argument: Please Enter number:53 Given number is prime with return and with argument: Given number is prime </pre>
Question	<p>1. You might be clear now, how user defined functions are created in different ways. Explain them.</p> <p>Ans:-</p> <p>Function with no arguments and no return value</p> <p>Such functions can either be used to display information or they are completely dependent on user inputs.</p>

Practical 9.2	<p>Verify the triangle, if the length of the sides of a triangle are denoted by a, b and c, then the area of triangle is given by:</p> $s = \frac{a+b+c}{2}$ $A = \sqrt{s(s-a) \times (s-b) \times (s-c)}$ <p>Use nested function. Collect the values for a, b and c from user for identifying whether it forms Triangle or not.</p>
Algorithm	<p>Step 1: Start Step 2: Input a, b, c Step 3: Calculate $s = (a + b + c) / 2$ Step 4: Calculate $\text{area} = \text{sqrt}(s*(s-a)*(s-b)*(s-c))$ Step 5: Print "Area of Triangle=", area Step 6: End</p>
Flow chart	 <pre> graph TD Start([Start]) --> Input[/Input a,b,c/] Input --> S["s=(a+b+c)/2"] S --> Area["area=sqrt((s*(s-a)*(s-b)*(s-c)))"] Area --> Print[/Print area/] Print --> End([End]) </pre>

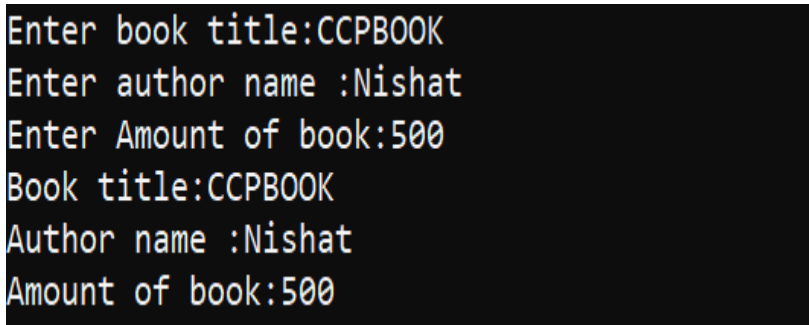
Code	<pre>#include<math.h> void length(int,int,int); int area(int,int,int,int); void main() { int a,b,c; printf("Enter value of a:"); scanf("%d",&a); printf("Enter value of b:"); scanf("%d",&b); printf("Enter value of c:"); scanf("%d",&c); length(a,b,c); } void length(int a,int b, int c){ if((a+b)>c&&(b+c)>a&&(c+a)>b&&a>0&&b>0&&c>0){ int ara,s; s=(a+b+c)/2; ara=area(a,b,c,s); printf("area of tringle: %d",ara);} else printf("The side of tringle is not vaild");} int area(int a,int b, int c,int s){ int ar =sqrt(s*(s-a)*(s-b)*(s-c)); return ar;}</pre>																											
Output Screenshot	<div><pre>Enter value of a:5 Enter value of b:12 Enter value of c:13 area of tringle: 30 Process returned 20 (0x14) execution time : 7.451 s Press any key to continue.</pre></div> <table><tr><th rowspan="2">Sr.No.</th><th colspan="3">Input</th><th rowspan="2">Forming a triangle</th><th rowspan="2">Not forming a triangle</th></tr><tr><th>a</th><th>b</th><th>c</th></tr><tr><td>1</td><td>5</td><td>12</td><td>13</td><td>✓</td><td></td></tr><tr><td>2</td><td>3</td><td>4</td><td>5</td><td>✓</td><td></td></tr><tr><td>3</td><td>1</td><td>1</td><td>3</td><td></td><td>✓</td></tr></table>	Sr.No.	Input			Forming a triangle	Not forming a triangle	a	b	c	1	5	12	13	✓		2	3	4	5	✓		3	1	1	3		✓
Sr.No.	Input			Forming a triangle	Not forming a triangle																							
	a	b	c																									
1	5	12	13	✓																								
2	3	4	5	✓																								
3	1	1	3		✓																							
Question	<p>Explain the concept of nested functions in C.</p> <p>Ans: A nested function is a function defined inside another function. Nested functions are supported as an extension in GNU C, but are not supported by GNU C++. The nested function's name is local to the block where it is defined</p>																											

Practical9.3	A positive integer is entered through the keyboard, write a function to find the binary equivalent of this number using recursion.
Algorithm	<p>Step 1: Start.</p> <p>Step 2: Input the number.</p> <p>Step 3: Divide the number by 2 through % (modulus operator) and store the remainder in array</p> <p>Step 4: Divide the number by 2 through / (division operator)</p> <p>Step 5: Repeat the step 4 until number is greater than 0</p> <p>Step 6: End</p>
Flow chart	 <pre> graph TD Start([Start]) --> Input[/Input number/] Input --> Calc[no%2 no/2] Calc --> Cond{no > 0} Cond -- True --> Calc Cond -- Text --> Print[/Print binary number/] Print --> End([End]) </pre>

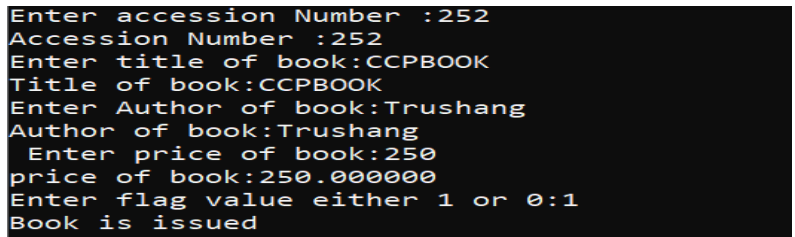
Code	<pre>void binary(int n); void main() { int n; printf("Enter positive integer:"); scanf("%d",&n); binary(n); } void binary(int n) { if(n!=0) { binary(n/2); printf("%d ",n%2); } }</pre>												
Output Screenshot	<div><pre>Enter positive integer:116 1 1 1 0 1 0 0 Process returned 2 (0x2) execution time : 6.504 s Press any key to continue.</pre></div> <div><pre>Enter positive integer:120 1 1 1 1 0 0 0 Process returned 2 (0x2) execution time : 3.365 s Press any key to continue.</pre></div> <div><pre>Enter positive integer:118 1 1 1 0 1 1 0 Process returned 2 (0x2) execution time : 5.196 s Press any key to continue.</pre></div> <table><thead><tr><th>Sr. No.</th><th>Input</th><th>Binary</th></tr></thead><tbody><tr><td>1</td><td>116</td><td>1 1 1 0 1 0 0</td></tr><tr><td>2</td><td>118</td><td>1 1 1 0 1 1 0</td></tr><tr><td>3</td><td>120</td><td>1 1 1 1 0 0 0</td></tr></tbody></table>	Sr. No.	Input	Binary	1	116	1 1 1 0 1 0 0	2	118	1 1 1 0 1 1 0	3	120	1 1 1 1 0 0 0
Sr. No.	Input	Binary											
1	116	1 1 1 0 1 0 0											
2	118	1 1 1 0 1 1 0											
3	120	1 1 1 1 0 0 0											
Question	<p>Mention the advantages of using recursion in a function.</p> <p>Ans:</p> <ol style="list-style-type: none">1. The code may be easier to write.2. To solve such problems which are naturally recursive such as tower of Hanoi.3. Reduce unnecessary calling of function.4. Extremely useful when applying the same solution.5. Recursion reduce the length of code.6. It is very useful in solving the data structure problem.7. Stacks evolutions and infix, prefix, postfix evaluations etc.												

Sign:**Grade:**

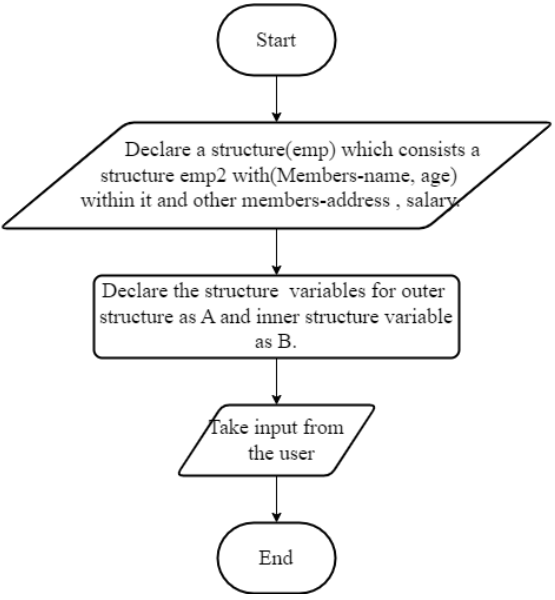
	SET:-10 Structures and Unions
Practical 10.1	<p>Write a C program to create a structure of Book Detail and display the details of the book in appropriate format by passing structure as a function argument.</p> <p>Book Detail must contain following information: Book Title, Author name and Amount of book in float.</p>
Algorithm	<p>Step 1: Start</p> <p>Step 2: Declare a structure (emp) which consists a structure emp2(With Members -name, age) within it and other members-address, salary.</p> <p>Step 3: Declare the structure variables for outer structure as A and inner structure variable as B.</p> <p>Step 4: Take input from user(Name of employee,age,address,and salary)and store it in Structure Variable with particular data type consisting member as per requirement.</p> <p>Step 5:End.</p>
Flow chart	<pre> graph TD Start([Start]) --> Declare[Declare structure with members holding Character array(string) for Book title and Author name, float for amount] Declare --> DeclareVar[Declare structure variable b1,b2,b3] DeclareVar --> Input[/Take input from user and store it in Structure variable ascending structure member/] Input --> End([End]) </pre>

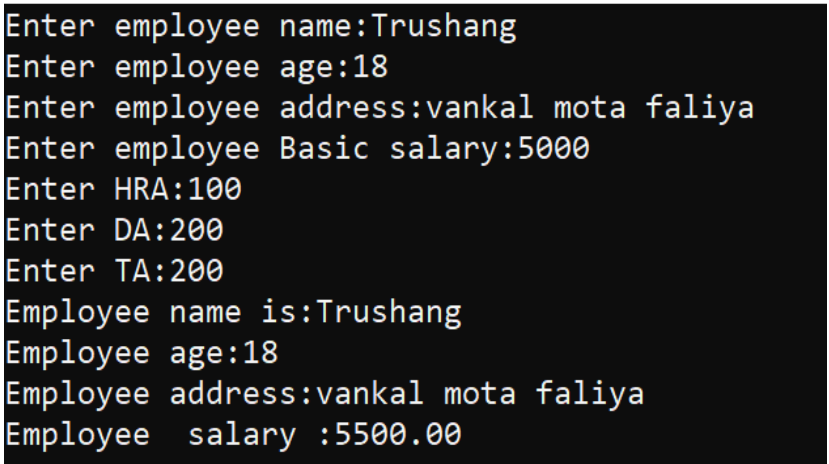
Code	<pre>struct book { char title[15]; char name[15]; int amt; }b; void f(struct book b) { printf("Enter book title:"); scanf("%s",b.title); printf("Enter author name :"); scanf("%s",b.name); printf("Enter Amount of book:"); scanf("%d",&b.amt); printf("Book title:%s\n",b.title); printf("Author name :%s\n",b.name); printf("Amount of book:%d\n",b.amt); } void main() { f(b); }</pre>												
Output Screenshot	<div></div> <table><tr><th>Sr . No</th><th>Book Title</th><th>Author name</th><th>Amount of book</th></tr><tr><td>1</td><td>1984</td><td>George</td><td>156.57</td></tr><tr><td>2</td><td>CCPBOOK</td><td>Nishat</td><td>500</td></tr></table>	Sr . No	Book Title	Author name	Amount of book	1	1984	George	156.57	2	CCPBOOK	Nishat	500
Sr . No	Book Title	Author name	Amount of book										
1	1984	George	156.57										
2	CCPBOOK	Nishat	500										
Question	<p>Can we declare function inside structure of C Programming? Explain Why?</p> <p>Ans: No, you cannot define a function inside the structure of C Programming, but you can do so in C++, rather you can have a function pointer in a “struct” in C Language.</p>												

Practical 10.2	Create a Union called library to hold accession number, title of the book, author name, price of the book and flag indicating whether the book is issued or not. (flag = 1 if the book is issued, flag = 0 otherwise). Write a program to enter data of one book and display the data.
Algorithm	<p>Step 1: Start</p> <p>Step 2: Declare a union with members holding character value ,float and integer value.</p> <p>Step 3: Declare union variables a1,a2,f.</p> <p>Step 4: Take input from the user and store it in union variables. Member according to requirement.</p> <p>Step 5: Check the flag using ' if ' condition(if flag is equal to 1 then book is issued).</p> <p>Step 6: End.</p>
Flow chart	<pre> graph TD Start([Start]) --> DeclareUnion[/Declare a union with members holding character value, float and integer value/] DeclareUnion --> DeclareVars[Declare union variables a1,a2,f.] DeclareVars --> TakeInput[/Take input from the user and store it in union variables, member according to requirement./] TakeInput --> Flag{Flag==1} Flag -- False --> NotIssued[/Book is not issued/] Flag -- True --> Issued[/Book is issued/] NotIssued --> End([End]) Issued --> End </pre>

Code	<pre> union library { int num; char title[15]; char au[15]; float price; int flag; }lib; void main() { printf("Enter accession Number :"); scanf("%d",&lib.num); printf("Accession Number :%d\n",lib.num); printf("Enter title of book:"); scanf("%s",lib.title); printf("Title of book:%s\n",lib.title); printf("Enter Author of book:"); scanf("%s",lib.au); printf("Author of book:%s\n",lib.au); printf(" Enter price of book:"); scanf("%f",&lib.price); printf("price of book:%f\n",lib.price); printf("Enter flag value either 1 or 0:"); scanf("%d",&lib.flag); if(lib.flag==1) printf("Book is issued"); else printf("Book is not issued"); } </pre>
Output Screenshot	 <pre> Enter accession Number :252 Accession Number :252 Enter title of book:CCPBOOK Title of book:CCPBOOK Enter Author of book:Trushang Author of book:Trushang Enter price of book:250 price of book:250.000000 Enter flag value either 1 or 0:1 Book is issued </pre>

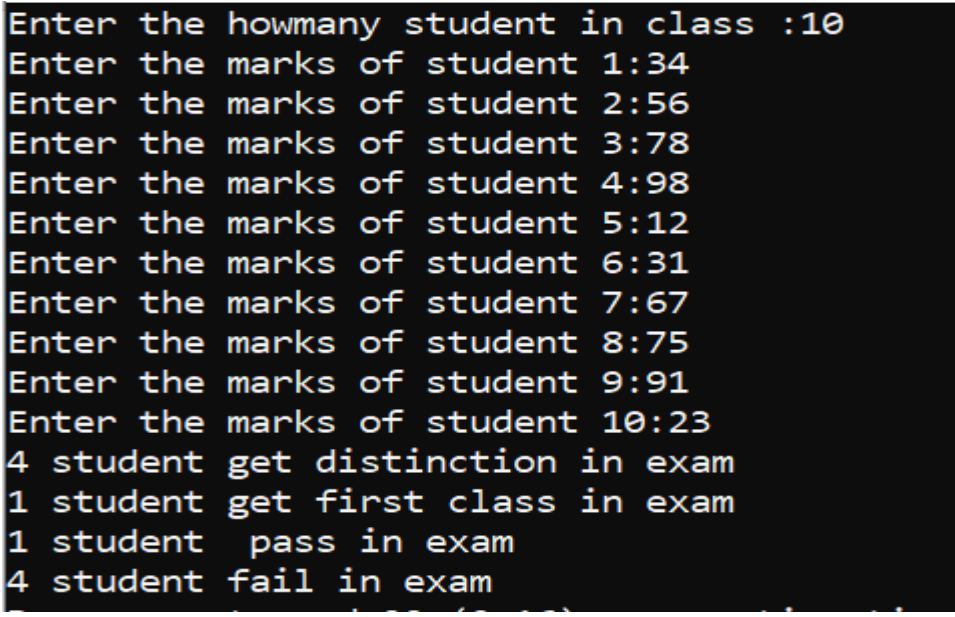
	Sr.No.	Accession Number	Title of book	Author	Price	Flag	Output
	1	10	Ansi C	Balaguru	250	1	Book issued
	2	8	BEEE	V.K. Mehta	175	0	Book not issued
	3	252	CCP	Trushang	250	1	Book issued
Question	<p>Explain the major difference between structure and union in detail.</p> <p>Ans: A structure is a user-defined data type available in C that allows to combining data items of different kinds. Structures are used to represent a record. A union is a special data type available in C that allows storing different data types in the same memory location.</p>						

Practical 10.3	Write a C program for collecting and displaying employee details such as, Age, Name, Address and Salary by using nested structure.
Algorithm	<p>Step 1: Start</p> <p>Step 2: Declare a structure (emp) which consists a structure emp2(With Members -name, age) within it and other members-address, salary.</p> <p>Step 3: Declare the structure variables for outer structure as A and inner structure variable as B.</p> <p>Step 4:Take input from user(Name of employee,age,address,and salary)and store it in Structure Variable with particular data type consisting member as per requirement.</p> <p>Step 5:End</p>
Flow chart	 <pre> graph TD Start([Start]) --> Decl1[/Declare a structure(emp) which consists a structure emp2 with(Members-name, age) within it and other members-address , salary/] Decl1 --> Decl2[Declare the structure variables for outer structure as A and inner structure variable as B.] Decl2 --> Input[/Take input from the user/] Input --> End([End]) </pre>
Code	<pre> struct book { char title[15] , name[15]; int amt; }b; void f(struct book b) </pre>

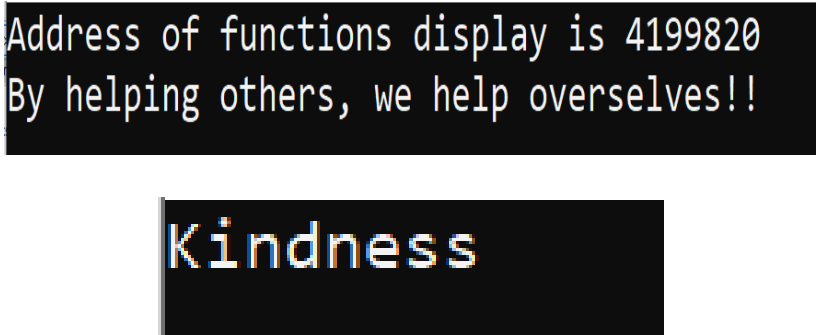
	<pre> { printf("Enter book title:"); scanf("%s",b.title); printf("Enter author name :"); scanf("%s",b.name); printf("Enter Amount of book:"); scanf("%d",&b.amt); printf("Book title:%s\n",b.title); printf("Author name :%s\n",b.name); printf("Amount of book:%d\n",b.amt); } void main() { f(b); } </pre>
Output	 <pre> Enter employee name:Trushang Enter employee age:18 Enter employee address:vankal mota faliya Enter employee Basic salary:5000 Enter HRA:100 Enter DA:200 Enter TA:200 Employee name is:Trushang Employee age:18 Employee address:vankal mota faliya Employee salary :5500.00 </pre>
Question	<p>Explain how nested structure works in C programming.</p> <p>Ans: A nested structure in C is a structure within structure. One structure can be declared inside another structure in the same way structure members are declared inside a structure.</p>

Sign:**Grade:**

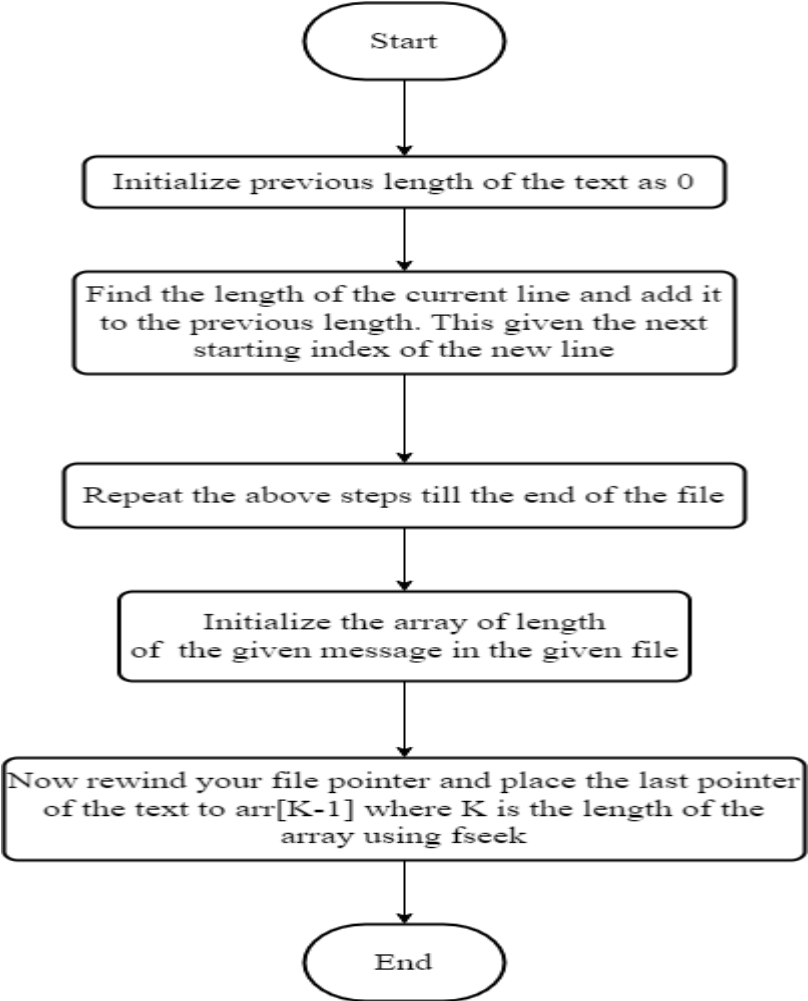
	SET:-11 Pointers
Practical 11.1	Write a program to read the marks of 10 students for the subject CE143 Computer concepts and Programming and compute the number of students in categories FAIL, PASS, FIRST CLASS and DISTINCTION using Pointers and Arrays.
Algorithm	<p>Step 1: Start</p> <p>Step 2: Declare marks[10],i=0, distinction=0,pass=0,fail=0,first_class=0.</p> <p>Step 3: Declare pointer array ptr_marks[10]</p> <p>Step 4: Input marks of 10 students from the user.</p> <p>Step 5: give address of marks to pointer ptr_marks.</p> <p>Step 6: If *ptr_marks[i]>=70 then distinction++ Else if (*ptr_marks[i]>=60&&*ptr_marks[i]<=69) then first_class++ Else if (*ptr_marks[i]>=40&&*ptr_marks[i]<=59) the pass++ Else fail++.</p> <p>Step 7: i++ If i<10 then goto step 6 else goto step 8.</p> <p>Step 8: Print value of distinction, first_class,fail,pass.</p> <p>Step 9: End.</p>
Flow chart	<pre> graph TD Start([Start]) --> Declare[/Declare a[10],n,d=0,fc=0,p=0 f=0,*x,i and take n ,x=a,x+i/] Declare --> Cond1{*(x+i)>=70 && *(x+i)<=100} Cond1 -- True --> dinc[d++] Cond1 -- False --> Cond2{*(x+i)>=60 && *(x+i)<=69} Cond2 -- True --> fcinc[fc++] Cond2 -- False --> Cond3{*(x+i)>=40 && *(x+i)<=59} Cond3 -- True --> pinc[p++] Cond3 -- False --> finc[f++] dinc --> Print[/Print d , fc, p,f/] fcinc --> Print pinc --> Print finc --> Print Print --> End([End]) </pre>

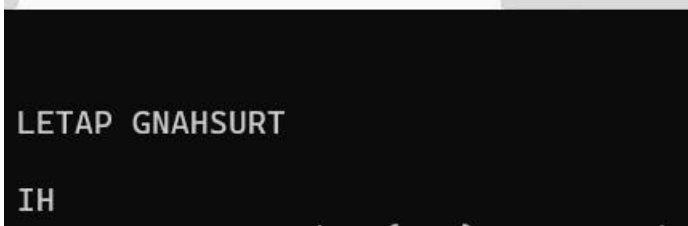
Code	<pre> void main() { int a[10],n,d=0,fc=0,p=0,f=0,*x,i; printf("Enter the howmany student in class :"); scanf("%d",&n); x=a; for(i=0;i<n;i++) { printf("Enter the marks of student %d:",i+1); scanf("%d",x+i); } for(i=0;i<n;i++) { if(*(x+i)>=70 && *(x+i)<=100) d++; else if(*(x+i)>=60 && *(x+i)<=69) fc++; else if(*(x+i)>=40 && *(x+i)<=59) p++; else f++; } printf("%d student get distinction in exam\n",d); printf("%d student get first class in exam\n",fc); printf("%d student pass in exam\n",p); printf("%d student fail in exam",f); } </pre>
Output Screenshot	 <pre> Enter the howmany student in class :10 Enter the marks of student 1:34 Enter the marks of student 2:56 Enter the marks of student 3:78 Enter the marks of student 4:98 Enter the marks of student 5:12 Enter the marks of student 6:31 Enter the marks of student 7:67 Enter the marks of student 8:75 Enter the marks of student 9:91 Enter the marks of student 10:23 4 student get distinction in exam 1 student get first class in exam 1 student pass in exam 4 student fail in exam </pre>

	<table><tr><th>Sr.No.</th><th>Input</th><th>Distinction</th><th>First class</th><th>Pass</th><th>Fail</th></tr><tr><td>1</td><td>56</td><td></td><td></td><td>✓</td><td></td></tr><tr><td>2</td><td>68</td><td></td><td>✓</td><td></td><td></td></tr><tr><td>3</td><td>75</td><td>✓</td><td></td><td></td><td></td></tr><tr><td>4</td><td>85</td><td>✓</td><td></td><td></td><td></td></tr><tr><td>5</td><td>98</td><td>✓</td><td></td><td></td><td></td></tr><tr><td>6</td><td>45</td><td></td><td></td><td>✓</td><td></td></tr><tr><td>7</td><td>36</td><td></td><td></td><td></td><td>✓</td></tr><tr><td>8</td><td>56</td><td></td><td></td><td>✓</td><td></td></tr><tr><td>9</td><td>57</td><td></td><td></td><td>✓</td><td></td></tr><tr><td>10</td><td>65</td><td></td><td>✓</td><td></td><td></td></tr><tr><td></td><td>Counts</td><td>3</td><td>2</td><td>4</td><td>1</td></tr></table>	Sr.No.	Input	Distinction	First class	Pass	Fail	1	56			✓		2	68		✓			3	75	✓				4	85	✓				5	98	✓				6	45			✓		7	36				✓	8	56			✓		9	57			✓		10	65		✓				Counts	3	2	4	1
Sr.No.	Input	Distinction	First class	Pass	Fail																																																																				
1	56			✓																																																																					
2	68		✓																																																																						
3	75	✓																																																																							
4	85	✓																																																																							
5	98	✓																																																																							
6	45			✓																																																																					
7	36				✓																																																																				
8	56			✓																																																																					
9	57			✓																																																																					
10	65		✓																																																																						
	Counts	3	2	4	1																																																																				
Question	<p>Explain the importance of using pointers.</p> <p>Ans: Importance of using pointers are as follows:</p> <ul style="list-style-type: none">(i) Pointers make the programs simple and reduce their length.(ii) Pointers are helpful in allocation and de-allocation of memory during the execution of the program. Thus, pointers are the instruments of dynamic memory management.(iii) Pointers enhance the execution speed of a program.																																																																								

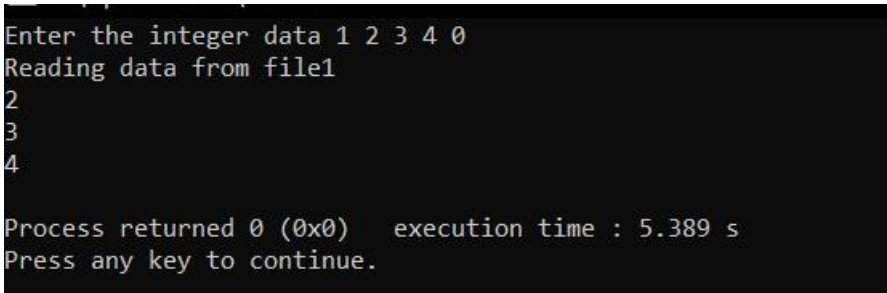
Practical 11.2	<p>Write output for the following programs:</p> <p>1.(Pointers to Functions) <pre>#include<stdio.h> void display(); int main(){ void (*func_ptr)(); func_ptr=display; printf("Address of functions display is %u\n",func_ptr); (*func_ptr)(); return 0;} void display() { puts("By helping others, we help overselves!!");}</pre></p> <p>2.(Functions returning pointers) <pre>char *copy (char*,char *); int main(){ char *str; char source[] = "Kindness"; char target[10]; str=copy(target,source); printf("%s\n",str); return 0;} char *copy(char *t,char *s){ char * r; r = t; while(*s!='\0'){ *t=*s; t++; s++; } *t='\0'; return(r);}</pre></p>
Output Screenshot	

Sign:**Grade:**

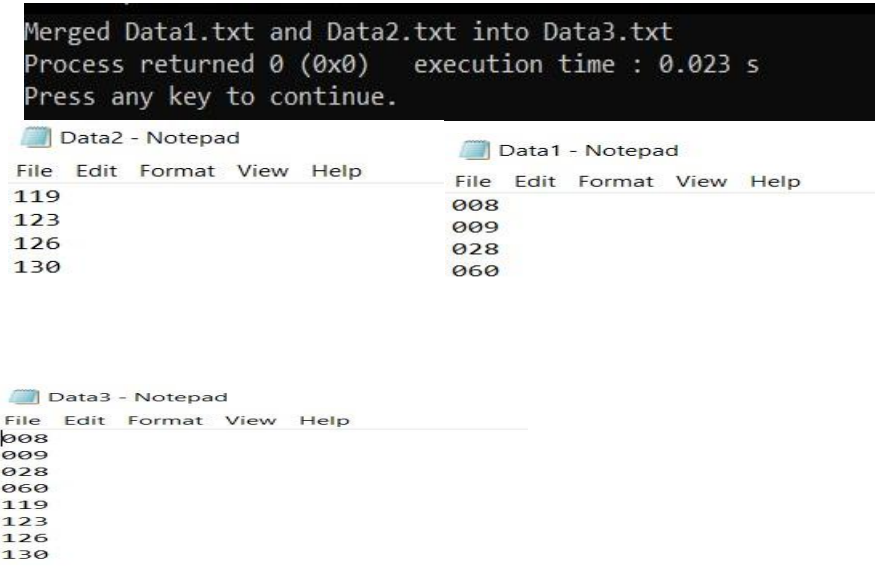
	SET-12 File Management in 'c'
Practical 12.1	Write a program to read a text file 'Demo.txt' and print each word of that file in reverse order.
Algorithm	<p>Step 1: Start</p> <p>Step 2: Initialise previous length of the text as 0.</p> <p>Step 3: Find the length of the current line and add it to the previous length. This given the next starting index of the new line.</p> <p>Step 4: Repeat the above steps till the end of the file.</p> <p>Step 5: Initialise the array of length of the given message in the given file.</p> <p>Step 6: Now rewind your file pointer and place the last pointer of the text to arr[K – 1] where K is the length of the array using fseek().</p> <p>Step 7: Print the length of the last line and decrease K by 1 for printing the next last line of the file.</p> <p>Step 8: Repeat the above steps untill K is equals to 0.</p> <p>Step 9: End.</p>
Flow chart	 <pre> graph TD Start([Start]) --> Init0[Initialize previous length of the text as 0] Init0 --> FindLen[Find the length of the current line and add it to the previous length. This given the next starting index of the new line] FindLen --> Repeat[Repeat the above steps till the end of the file] Repeat --> InitArr[Initialize the array of length of the given message in the given file] InitArr --> Rewind[Now rewind your file pointer and place the last pointer of the text to arr[K-1] where K is the length of the array using fseek] Rewind --> End([End]) </pre>

Code	<pre> #include<stdio.h> int main() { FILE *fp; char ch; int i,pos; fp=fopen("Demo.txt","r"); if(fp==NULL) { printf("File does not exist.."); } fseek(fp,0,SEEK_END); pos=ftell(fp); //printf("Current position is %d\n",pos); i=0; while(i<pos) { i++; fseek(fp,-i,SEEK_END); //printf("%c",fgetc(fp)); ch=fgetc(fp); printf("%c",ch); } return 0; } </pre>
Output Screenshot	
Question	<p>Explain why do we need to use files in C?</p> <p>Ans: Entire data is lost when the program terminates and storing in a file will preserve your data even if the program terminates. ... If you have a file containing all the data, you can easily access the contents of the file by using few commands in C.</p>

Practical 12.2	Write a C program that illustrates how to write into a file using putw() function and how to read the same file using getw() function.
Algorithm	Step 1: Start Step 2: Create the file. Step 3: Read the file using getw() function. Step 4: Print the file using putw() function. Step 5: End
Flow chart	<pre>graph TD; Start([Start]) --> Create[/Create the file/]; Create --> Read[Read the file using getw() function]; Read --> Print[/Print the file using putw() function/]; Print --> End([End]);</pre>

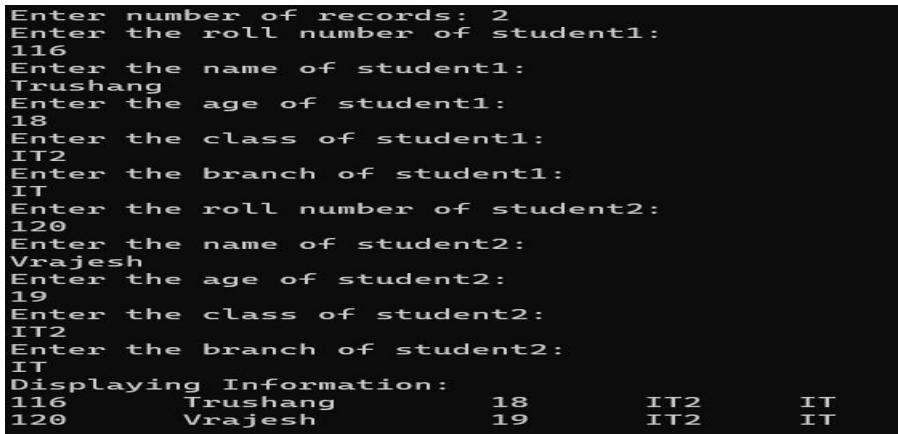
Code	<pre>#include<stdio.h> int main() { FILE *fp; int n; fp=fopen("c.dat", "wb+"); printf("Enter the integer data"); scanf("%d",&n); while(n!=0) { putw(n,fp); scanf("%d",&n); } rewind(fp); printf("Reading data from file"); while((n=getw(fp))!=EOF) { printf("%d\n",n); } fclose(fp); return 0; }</pre>												
Output Screenshot													
Question	<p>Explain any 3 functions of file other then mentioned in the program</p> <table><tr><th>Sr.No.</th><th>Function</th><th>Purpose</th></tr><tr><td>1</td><td>fgetc()</td><td>fgetc () function reads a character from file.</td></tr><tr><td>2</td><td>fscanf()</td><td>fscanf () function reads formatted data from a file.</td></tr><tr><td>3</td><td>ftell()</td><td>ftell () function gives current position of file pointer.</td></tr></table>	Sr.No.	Function	Purpose	1	fgetc()	fgetc () function reads a character from file.	2	fscanf()	fscanf () function reads formatted data from a file.	3	ftell()	ftell () function gives current position of file pointer.
Sr.No.	Function	Purpose											
1	fgetc()	fgetc () function reads a character from file.											
2	fscanf()	fscanf () function reads formatted data from a file.											
3	ftell()	ftell () function gives current position of file pointer.											

Practical 12.3	Two files Data1.txt and Data2.txt contains list of integers. Write a program to produce file Data3.txt which holds as merged list of these two lists. Use command line argument to specify the file name.
Algorithm	Step 1: Start. Step 2: Input Integers in Data1.txt and Data2.txt. Step 3: Merge integers entered in Data1.txt and Data2.txt into Data3.txt. Step 4: Print Merged Data1.txt and Data2.txt into Data3.txt. Step 5: End
Flow chart	<pre>graph TD; Start([Start]) --> Input[/Input Integers in Data1.txt and Data2.txt/]; Input --> Merge[Merge integers entered in Data1.txt and Data2.txt]; Merge --> Print[/Print Merged Data1.txt and Data2.txt into Data3.txt/]; Print --> End([End]);</pre>

Code	<pre> #include <stdio.h> #include <stdlib.h> int main() { FILE *fp1 = fopen("Data1.txt", "r"); FILE *fp2 = fopen("Data2.txt", "r"); FILE *fp3 = fopen("Data3.txt", "w"); char c; if (fp1 == NULL fp2 == NULL fp3 == NULL) { puts("Could not open files"); exit(0); } while ((c = fgetc(fp1)) != EOF) fputc(c, fp3); while ((c = fgetc(fp2)) != EOF) fputc(c, fp3); printf("Merged Data1.txt and Data2.txt into Data3.txt"); fclose(fp1); fclose(fp2); fclose(fp3); return 0; } </pre>
Output Screenshot	
Question	<p>Explain the difference between argc and argv along with their significance.</p> <p>Ans: argc is the number of arguments being passed into your program from the command line and argv is the array of arguments.</p>

Sign:**Grade:**

	SET:-13 Dynamic Memory Allocation
Practical 13.1	<p>Write a program to read and print the student details using structure and Dynamic Memory Allocation.</p> <p>Following student details needs to be included: Roll No., Name, Age, Class, Branch.</p>
Algorithm	<p>Step 1: START</p> <p>Step 2: Declare struct student_details int Roll_no & age, char name[100], Classa[10] & branch[50]</p> <p>Step 3: Declare struct student_details *strdet, int i & n</p> <p>Step 4: INPUT The number of students</p> <p>Step 5: strdet=(struct student_details *)malloc(n*sizeof(struct student_details)) & i=0</p> <p>Step 6: If i<n, true then goto Step 7 and false then goto Step 9</p> <p>Step 7: INPUT The values of Roll.No., name, age, class and branch for (strdet+i)</p> <p>Step 8: i++, then goto Step 6</p> <p>Step 9: i=0</p> <p>Step 10: If i<n, true then goto Step 11 and false then goto Step 13</p> <p>Step 11: PRINT The values of Roll.No., name, age, class & branch for (strdet+i)</p> <p>Step 12: i++, then goto Step 10</p> <p>Step 13: END</p>
Flow chart	<pre> graph TD Start([Start]) --> Decl1[/Declare Struct student details, int Roll_no & age, Char name[100], class a[10] , & branch[50]/] Decl1 --> Decl2[Declare struct student_details , strdet, int i & n] Decl2 --> Input1[/Input the number of student and details/] Input1 --> Init[Strdet=(struct student_details*)malloc(n*sizeof(struct_details)) & i=0] Init --> If1{If} If1 -- True --> Input2[/Input values of roll no, name ,class & branch , age or (strdet++i)/] Input2 --> Inc1[i++] Inc1 --> If1 If1 -- False --> Init2[i=0] Init2 --> If2{If i<n} If2 -- True --> Print[/Print the value of Roll no.,name,class,age branch for(strdet++i)/] Print --> Inc2[i=0] Inc2 --> If2 If2 -- False --> End([End]) </pre>

Code	<pre> #include <stdio.h> #include<stdlib.h> struct information { int roll_number; int age; char class_[5]; char name[30]; char branch[30]; }; int main(){ struct information *ptr; int i, noOfRecords; printf("Enter number of records: "); scanf("%d", &noOfRecords); ptr = (struct information*) malloc (noOfRecords * sizeof(struct information)); for(i = 0; i < noOfRecords; ++i){ printf("Enter the roll number of student%d: \n",i+1); scanf("%d", &(ptr+i)->roll_number); printf("Enter the name of student%d: \n",i+1); scanf("%s", &(ptr+i)->name); printf("Enter the age of student%d: \n",i+1); scanf("%d", &(ptr+i)->age); printf("Enter the class of student%d: \n",i+1); scanf("%s", &(ptr+i)->class_); printf("Enter the branch of student%d: \n",i+1); scanf("%s", &(ptr+i)->branch);} printf("Displaying Information:\n"); for(i = 0; i < noOfRecords ; ++i) printf("%d\t%s\t%d\t%s\t%s\n", (ptr+i)->roll_number, (ptr+i)- >name,(ptr+i)->age,(ptr+i)->class_,(ptr+i)->branch); return 0; } </pre>
Output Screenshot	 <pre> Enter number of records: 2 Enter the roll number of student1: 116 Enter the name of student1: Trushang Enter the age of student1: 18 Enter the class of student1: IT2 Enter the branch of student1: IT Enter the roll number of student2: 120 Enter the name of student2: Vrajesh Enter the age of student2: 19 Enter the class of student2: IT2 Enter the branch of student2: IT Displaying Information: 116 Trushang 18 IT2 IT 120 Vrajesh 19 IT2 IT </pre>

	<table><tr><th>Sr.No.</th><th>Roll No.</th><th>Name</th><th>Age</th><th>Class</th><th>Branch</th></tr><tr><td>1</td><td>116</td><td>Trushang</td><td>18</td><td>IT2</td><td>IT</td></tr><tr><td>2</td><td>120</td><td>Vrajesh</td><td>19</td><td>IT2</td><td>IT</td></tr></table>	Sr.No.	Roll No.	Name	Age	Class	Branch	1	116	Trushang	18	IT2	IT	2	120	Vrajesh	19	IT2	IT
Sr.No.	Roll No.	Name	Age	Class	Branch														
1	116	Trushang	18	IT2	IT														
2	120	Vrajesh	19	IT2	IT														
Question	<p>Explain the benefits of using dynamic memory allocation. Give one scenario where it is most useful.</p> <p>Ans:- Benefits of Dynamic memory allocation are</p> <ul style="list-style-type: none">• Data structures can grow and shrink according to the requirement.• We can allocate additional storage whenever we need them.• We can de-allocate dynamic space whenever we are done with them.• Dynamic Allocation is done at run time. <p>For example, the list of students in one department in a college. This dynamic memory allocation is most useful, because one can leave the college and one can join the college, so, according to these the memory allocation may increase or decrease.</p>																		

Practical 13.2

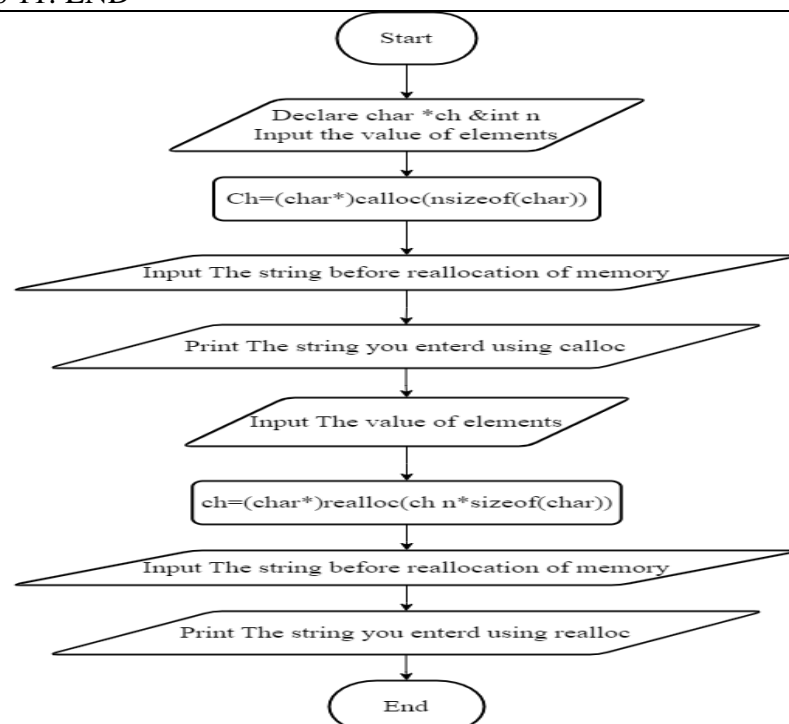
Write a program using a character string in a block of memory space created by calloc() and then modify the same to store a larger string using realloc () function. (Dynamic Array).

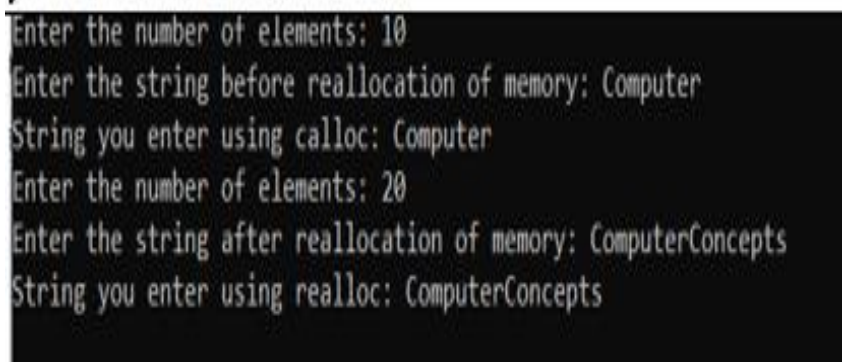
Expected Outcome:

SR NO.	INSTRUCTION	OUTPUT
1	String to be entered	
2	String received after reallocation of memory	

Algorithm

Step 1: START
 Step 2: Declare char *ch
 and int n
 Step 3: INPUT The value of
 elements
 Step 4: ch=(char *)calloc(n,sizeof(char))
 Step 5: INPUT The string before the reallocation of
 memory
 Step 6: PRINT The string you enter using calloc
 Step 7: INPUT The value of elements
 Step 8: ch=(char *)realloc(ch,n*sizeof(char))
 Step 9: INPUT The string after reallocation of of memory
 Step 10: PRINT The string you enter using realloc
 Step 11: END

Flow chart

Code	<pre>#include<stdlib.h> void main() { Char *ch; int n; printf("Enter the number of elements:"); scanf("%d",&n); ch=(char *)calloc(n,sizeof(char)); printf("Enter the string before reallocation of memory:"); scanf("%s",ch); printf("String you enter using calloc: %s",ch); printf("\nEnter the number of elements:"); scanf("%d",&n); ch=(char *)realloc(ch,n*sizeof(char)); printf("Enter the string after reallocation of memory:"); scanf("%s",ch); printf("String you enter using realloc: %s",ch); }</pre>									
Output Screenshot	<div></div> <table><tr><th>Sr.No.</th><th>Instruction</th><th>Output</th></tr><tr><td>1</td><td>String to be entered</td><td>Computer</td></tr><tr><td>2</td><td>String received after reallocation of memory</td><td>ComputerConcepts</td></tr></table>	Sr.No.	Instruction	Output	1	String to be entered	Computer	2	String received after reallocation of memory	ComputerConcepts
Sr.No.	Instruction	Output								
1	String to be entered	Computer								
2	String received after reallocation of memory	ComputerConcepts								
Question	<p>Mention advantage of using realloc() function.</p> <p>Ans:- Advantage of using realloc() function is it's ~4 times faster than malloc()/free() and copying your data when scaling up, when scaling down it is 10000-100000 times faster. Never copy stuff manually.</p>									

Practical 13.3

Write a program to enter N numbers into array and find average. Enter the size of the array through keyboard. (Dynamic Array). Use malloc () to allocate memory and use free() to free the memory after the use.

Expected Outcome:

Average of entered values		To be entered by user
...		To be entered by user
2.		To be entered by user
1.		To be entered by user
Enter the size of Array		N (To be entered by user)
St.No.	Instruction	Output

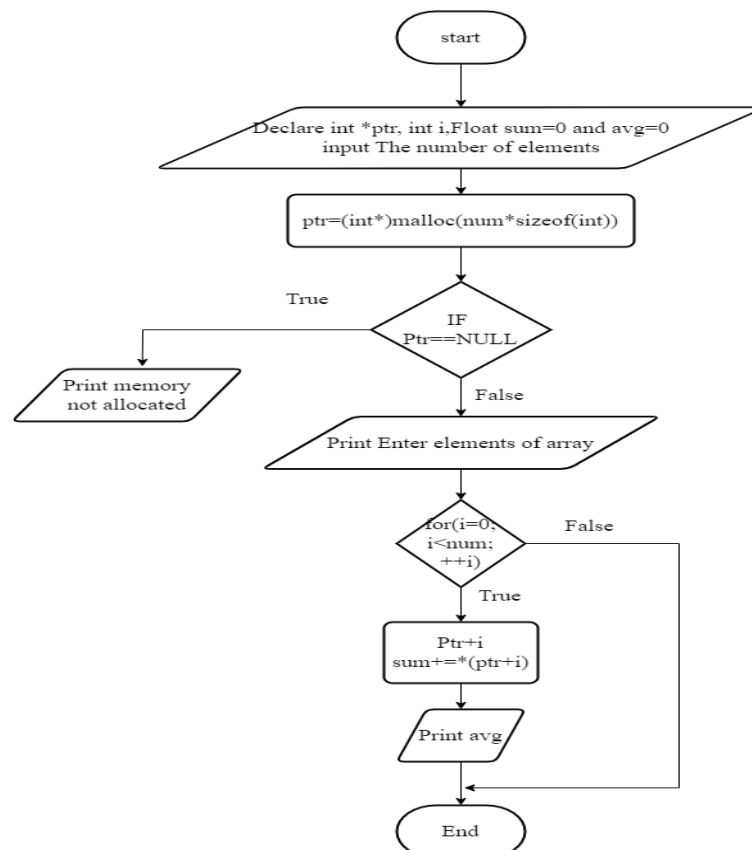
Algorithm

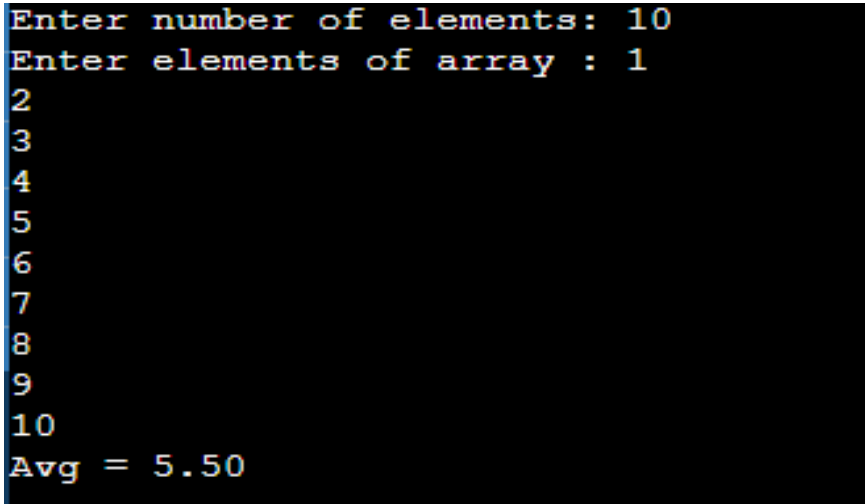
```

Step 1: START
Step 2: Declare int I and
* ptr
Step 3:- declare float
sum=0 and avg=0.
Step 4: ptr = (int*) malloc(num * sizeof(int))
Step 5: PRINT The 10 interger enter using malloc
Step 6: check if ptr==null
Step 7: True then file not found
Step 8: then for ptr +i
    sum += *(ptr + i)
Step 9: printf("Avg = %0.2f", sum/10)
Step 10:- free(ptr)
Step 11:- Stop.

```

Flow chart



Code	<pre>#include <stdio.h> #include <stdlib.h> int main() { int num, i, *ptr; float sum = 0,Avg=0; printf("Enter number of elements: "); scanf("%d", &num); ptr = (int*) malloc(num * sizeof(int)); if(ptr == NULL) { printf("Error! memory not allocated."); exit(0); } printf("Enter elements of array: "); for(i = 0; i < num; ++i) { scanf("%d", ptr + i); sum += *(ptr + i); } printf("Avg = %0.2f", sum/10); free(ptr); return 0; }</pre>
Output Screenshot	 <p>The screenshot shows the execution of the C program. It prompts the user to enter the number of elements, which is 10. Then it prompts for the elements of the array, which are 1 through 10. Finally, it displays the calculated average as 5.50.</p>

Question			
	Sr.No.	Instruction	Output
	Enter the size of Array		10
	1		1
	2		2
	3		3
	4		4
	5		5
	6		6
	7		7
	8		8
	9		9
	10		10
	Average of the entered values		5.50

Sign:**Grade:**